

Internet Security and Acceleration Server

Bonus CHAPTER

IN THIS CHAPTER

- **ISA Server Capabilities 3**
- **Managing ISA Server 43**
- **Configuring ISA Server Clients 116**
- **Safe and (More) Secure 125**

You probably already realize that connecting your organization's network to the Internet without a firewall is like driving with bad brakes—something disastrous is bound to happen eventually. Read my book *Drew Heywood's Windows 2000 Network Services* if you need convincing. RRAS includes a network address translating firewall. NAT is an effective firewall technique, but there are some desirable capabilities that NAT lacks.

Here are a few NAT improvements that could be suggested:

- NAT uses one security policy that applies to all users communicating through the firewall. It would be very useful to be able to apply different policies to different users or groups.
- NAT has no method of authenticating users. If we want to securely control access to resources, however, an authentication mechanism is required.
- NAT cannot limit access to specific Internet resources. Users can obtain stock quotes and dirty pictures just as easily as they can get technical information on `Microsoft.com`. Unless recreational browsing is in our company business plans, we would like some control over the Internet resources users can access.
- NAT doesn't raise any red flags when intruders try to vaporize your network. We would like to know when an unwanted guest is knocking on the door.

Enter the Microsoft Internet Security and Acceleration Server (ISA Server), which supplements the limited firewall capabilities of RRAS in a variety of ways. (In case you're suffering from acronym overload, and who isn't at this point, don't confuse ISA with IAS, the Internet Authentication Server discussed in Chapter 8 of the book.) ISA's firewall component controls outgoing network traffic using policies that specify site, content, and protocol restrictions. Policies can be applied depending on the characteristics of network traffic or on the identities of users and their group memberships. Services on local computers can be made available to the outside without compromising security.

As its full name suggests, security isn't ISA's only capability, but how does ISA "accelerate" Internet access? Apart from blowing the budget on ever-faster Internet connections and Web servers, caching is the most powerful tool available for improving data access performance. ISA also includes a caching component of ISA that can speed access to commonly used Web objects by retaining local copies. Often-used Web objects can be retrieved locally, improving responsiveness while promoting efficient use of WAN bandwidth. The result is a more efficient Web access environment that is less easily overloaded as demand increases.

NOTE

As I am writing, ISA is in pre-release testing. This chapter was prepared using ISA Release Candidate 1.

ISA Server Capabilities

ISA is much more complex than its predecessor (Proxy Server 2.0) and includes two major functions:

- Enabling internal users to access external services under administrative control that can be tailored to the needs of the organization.
- Blocking external access to internal computers while, if desired, enabling external clients to access select internal servers and services.

As an ISA Server administrator, you can allow or block any or all types of outbound and inbound packets, making the firewall as permeable as you require. To reduce the likelihood of inadvertent security leaks, ISA Server blocks all types of traffic that you do not explicitly allow. You must explicitly allow ISA Server to forward packets with specific characteristics.

ISA Server supports Windows and non-Windows clients, although different capabilities are available. For example, Windows clients can be authenticated using Windows-integrated authentication and can use a Windows-specific firewall. But UNIX, Linux, Macintosh, and other clients can still communicate through ISA Server. Access for non-Windows clients can even be identified using plain text or certificate-based authentication.

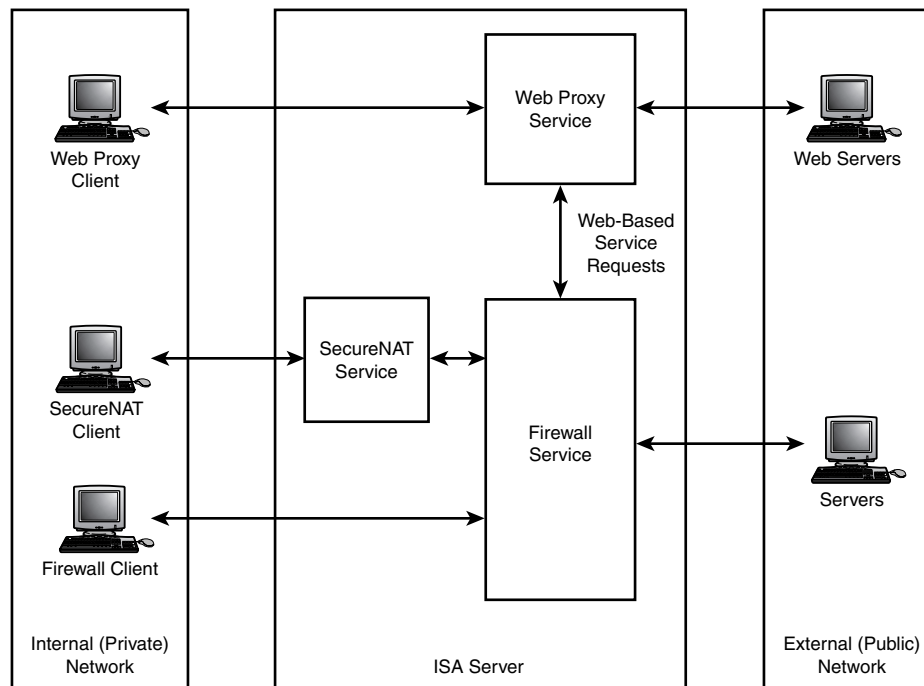
To prepare ourselves for ISA Server administration, we will first examine the components and characteristics of ISA Server.

Client Types

ISA Server provides firewall services for three types of clients:

- *Web Proxy clients* are applications such as Web browsers that support the CERN proxy protocol.
- *Windows firewall clients* are Windows clients running Windows Firewall Client software, enabling ISA to authenticate clients using Windows-integrated authentication and to control access using policies.
- *SecureNAT clients* are non-Windows computers or Windows computers not running the Windows Firewall Client. SecureNAT improves on the native Windows 2000 NAT with improved access filtering. We examined the Windows 2000 NAT in Chapter 7, “Routing with Routing and Remote Access Service.”

Each of these client types is supported by a matching service on the ISA server. The relationships among ISA clients and services are illustrated in Figure 1.

**FIGURE 1**

ISA supports Web Proxy, firewall, and SecureNAT clients with corresponding services.

- *Web Proxy clients* (usually Web browsers) are configured to direct all requests for outside Web resources to the Web Proxy service. The chief contribution of the Web Proxy service is the implementation of a cache that stores recently retrieved and frequently retrieved Web objects locally so that they can be served to clients without requesting them through the WAN. Any client software that supports the CERN proxy protocol can be a Web Proxy client, including non-Windows computers running a suitable Web browser.
- *Windows firewall clients* direct all requests for outside resources to the firewall service. Redirection is performed by a modified version of WinSock that determines whether service requests should be sent to internal servers or to ISA Server, which controls access to outside services. The firewall service can authenticate clients and use policies to determine whether the request is allowed. Requests for Web resources are redirected to the Web Proxy service. Thus all Windows firewall clients indirectly are Web Proxy clients. (Web Proxy redirection can be disabled.)

- *SecureNAT clients* are configured to use ISA simply by specifying the ISA server as the client's default router used to access outside resources. The SecureNAT service maps clients' internal IP addresses and ports to public addresses and ports, editing datagrams as necessary using the NAT component of RRAS. SecureNAT is a Windows firewall client, and edited datagrams are directed to the Windows firewall service, which applies its own access filters and directs requests to the WAN or to the Web Proxy service as appropriate.

A client cannot be both a firewall client and a SecureNAT client. Windows computers running the Windows Firewall Client software function as firewall clients. Windows computers not running Windows Firewall Client software—as well as UNIX, Linux, Macintosh, and other computers—function as SecureNAT clients.

However, both Windows firewall clients and SecureNAT clients can also be Web Proxy clients. In fact, with the default configuration of ISA Server, Windows firewall service is a Web Proxy client that directs requests for Web resources to the Web Proxy service.

NOTE

Before sending a service request, firewall Web Proxy clients must be able to determine whether the target server is internal or external so that they can decide whether to address the packets to a local server or to ISA Server. SecureNAT clients make this determination using their routing tables. Firewall and Web Proxy clients use a different mechanism.

Firewall clients make the local-remote determination by consulting a Local Address Table (LAT) that lists all IP address ranges that are to be regarded as internal. If a firewall client needs to direct a request to an IP address that isn't included in the LAT, the request must be sent to the ISA server. The LAT is maintained on ISA Server and is downloaded periodically to the client. ISA Server also supports a Local Domain Table (LDT) that serves a function similar to the LAT, listing domain names that are used on the local network.

Web Proxy clients make use of a local address table also, but don't use the LAT or LDT tables that are defined for the firewall service. Separate local address tables are configured in the ISA Server administration console or on individual Web Proxy clients.

The Web Proxy Service

In human terms, a proxy is “a person authorized to act as a substitute.” The Web Proxy server acts as a substitute for one or many network clients, letting them communicate with servers on another network without actually being openly connected to the remote network. It's a sleight-of-hand trick that provides firewall protection and some other benefits as well.

In part, a proxy server is a heavy-duty translator that acts as an intermediary between your network clients and the Internet. Look at Figure 2. Superficially, the proxy server looks like a router, but it isn't. A router forwards packets more or less intact from one network to another, extending the reach of the computer that originated the packet. A router has a tough, busy life, but it is mostly one of receiving packets and forwarding them to the correct network.

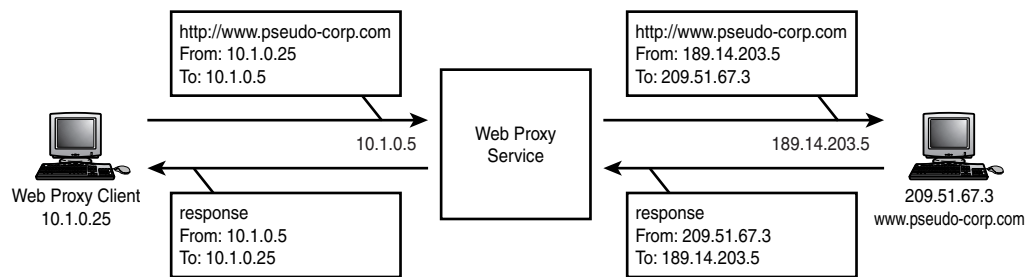


FIGURE 2

Operation of a proxy server.

But nothing is simply passed through a proxy server. In Figure 2, the client can't actually communicate with the Internet. It communicates with the proxy server *and thinks the proxy server is the Internet*. Similarly, servers on the Internet can't communicate with the clients on the private network. Instead, servers communicate with the proxy server *and think the proxy server is the client*. In between, the proxy server copies, translates, and forwards as required to facilitate communication. As such, the proxy server functions as a firewall between the private and public networks, permitting packets to enter the private network only if they are generated in response to requests from local clients.

A Web Proxy client is configured to direct requests for external Web services to a Web Proxy server. By default, the Web Proxy server accepts requests for HTTP, HTTPS, FTP, and Gopher objects on port 8080. Upon receiving a Web service request from a Web Proxy client, the Web Proxy server generates a new request that is sent to the Web server specified by the client. These requests are sent to the port assigned to the Web server service, for example, port 80 for HTTP.

It is important to realize how the behavior in Figure 2 differs from behavior if a router functioned in place of the Web Proxy service. If the middle box functioned as a router, the addressing shown for the service requests and service replies would be different. The client would resolve `www.pseudo-corp.com` to 209.51.67.3, and the service request would be addressed from 10.1.0.25 to 209.51.67.3. Recall that a router does not modify IP addresses as it forwards packets. Another difference is that the Web Proxy client directs all requests for external Web services to the Web Proxy. Thus, in a Web Proxy scenario, the client directs the Web service request to 10.1.0.3, not to the actual address of the Web server.

Although the World Wide Web emphasizes HTTP, several other protocols are also in common use in Web communication. The Web Proxy service supports the following protocols:

- HTTP
- FTP
- Gopher
- Secure HTTP (HTTPS; HTTP with Secure Sockets Layer)

Web Proxy Caching

The Web Proxy service can enhance Web communication by maintaining a cache of recently or frequently retrieved objects. When a Web Proxy client requests an object, the proxy server examines its cache before sending a query to the Internet. If the object is in the local cache, ISA Server can return it to the user without requesting it from the Internet. Consider the clients and Web Proxy service shown in Figure 3. The following events might take place:

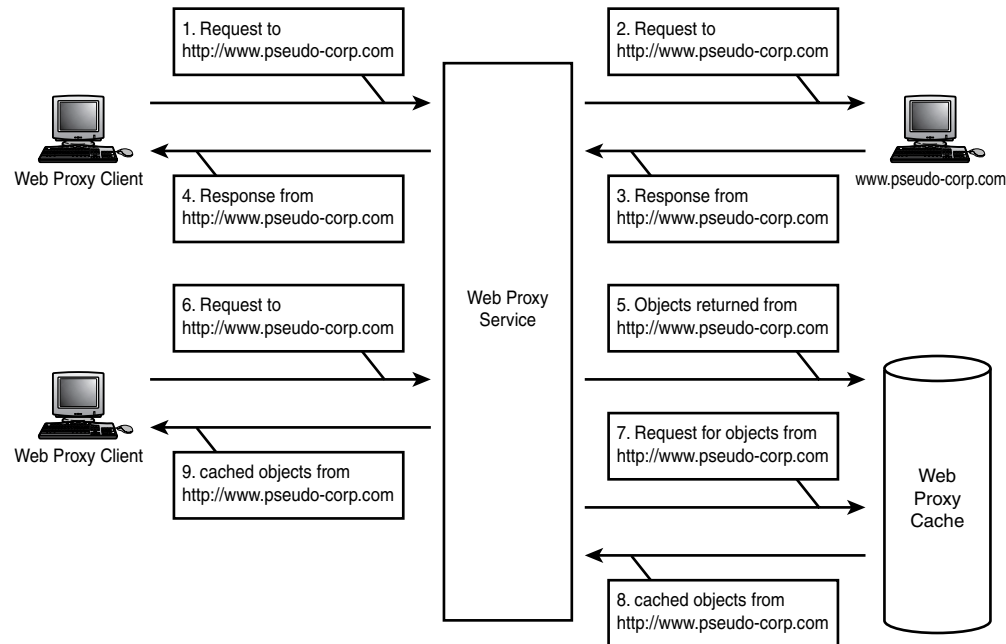
1. Web Proxy Client A requests the Web page `www.pseudo-corp.com`, directing the request to the Web Proxy server.
2. The Web Proxy requests `www.pseudo-corp.com`.
3. The Web Proxy receives the default HTML page from `www.pseudo-corp.com` along with any graphic, audio, or other objects that the page requires.
4. The Web Proxy forwards the objects to Client A.
5. If the objects are suitable for caching, the Web Proxy also stores them in its cache.
6. Web Proxy Client B requests the Web page `www.pseudo-corp.com`, directing the request to the Web Proxy server.
7. The Web Proxy examines its cache and, if the required objects are found, requests them from the cache.
8. The cache returns the objects to the Web Proxy server.
9. The Web Proxy service forwards the objects to Web Proxy Client B.

After Client A's request is fulfilled, subsequent requests for the same object can be satisfied locally. Clients receive the objects more quickly and there is no need to use WAN bandwidth to repeat the request.

ISA Server supports *scheduled caching*, enabling an administrator to specify Web sites that are to be cached on a periodic basis. Frequently used Web sites can be cached during periods of light traffic so that they are available for local access.

Forward and Reverse Caching

Caching can be configured to operate in forward and reverse modes. Forward caching, illustrated in Figure 3, stores external Web objects that are requested by internal users.

**FIGURE 3**

Caching is managed by the Web Proxy service.

Reverse caching, shown in Figure 4, is one of the techniques that has kept the World Wide Web from grinding to a halt. Reverse caching improves the efficiency and responsiveness of your Web servers when they provide objects to clients. It's one of the most effective ways to improve Web server performance without endlessly upgrading Web server hardware or expanding Web server farms.

Suppose that your organization operates a Web server that is publicly available to users of the Internet. Ordinarily, Web servers perform a lot of disk access. Every time a user requests an object from your Web server, the Web server must retrieve the object from disk. Think about your organization's home page for a moment. In most cases, every user retrieves the same objects when they connect to your site; as a result every new user connection results in disk activity to retrieve the exact same data. Also, cached objects can be served to clients without requiring retrieval from disk.

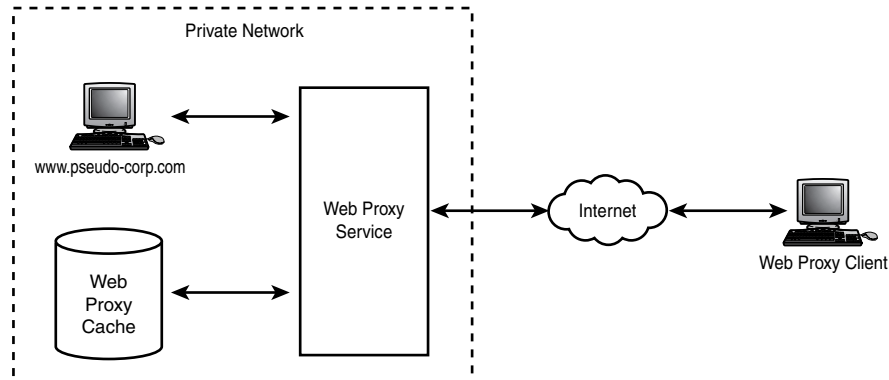


FIGURE 4
Reverse proxy caching.

Limitations of Caching

Caching is not a universal balm for Web performance ills because many Web objects are not suitable for caching. Suppose that you visit a Web site that presents a custom home page to each user, perhaps based on identity information stored in a cookie on the user's computer. Or suppose that a Web page is generated as a result of a query. These dynamically generated pages are of value only to a single user and then only for a short period of time. Clearly, dynamically generated Web pages such as these won't benefit from caching. When a Web server returns an object, it may specify a Time to Live for the object that declares the maximum amount of time the object should be held in cache. Otherwise, a default TTL can be configured for the cache.

Web Proxy Versus NAT

Superficially, the Web Proxy service looks like NAT. If you look at packets on the wire, however, significant differences are evident. Recall from Chapter 2, "TCP/IP Protocol Concepts," that a connection is established and closed with two explicit dialogs:

- A SYN/SYN-ACK/ACK dialog is used when the client negotiates a connection with a server.
- When it is time to close the connection, the client and server each initiate a FIN/FIN ACK dialog that flushes any untransmitted data and closes the connection in an orderly manner.

When a client communicates through a NAT, the connection setup and closing dialogs take place between the client and the outside server. The NAT translates the client address in its role as intermediary and may edit some internal packet details, but the client forms no TCP connections with the NAT firewall. The standard connection setup and teardown dialogs mentioned

above continue to apply. With the Web Proxy, however, the connection works differently. Specifically:

- The client opens a connection with the Web Proxy server, which *is* the Web server as far as the client is concerned.
- The Web Proxy server opens a separate connection with the Web server. As far as the Web server is concerned, the Web Proxy service *is* the client.

If the client requests an object that is in the Web Proxy server's cache, only the connection between the client and the Web Proxy server is required. That connection enables the Web Proxy to masquerade as the Web server and return the cached object to the client. As far as the client is concerned, it communicated with the Web server, but the Web server may not have any part in servicing the request.

ISA Server Arrays

Suppose that you manage the network of a large organization that generates too much traffic with external servers to be handled by a single ISA Server. You could set up multiple ISA Servers and configure groups of clients to use different ones, but that could be an administrative nightmare.

To promote scaling, ISA servers can be configured in *arrays* as shown in Figure 5. The array is named ISA Array 1. The array consists of three ISA Servers, identified as ISAServer1A, ISAServer1B, and ISAServer1C. ISA Servers in an array are configured from the same array policy, although a few properties can be configured independently on array members.

There are several advantages to ISA arrays:

- Services in the ISA Servers operate. Firewall clients can access any ISA Server in an array.
- Servers in an array are managed as a single entity and typically all servers are configured identically through the same properties.
- Web Proxy servers in an array share a virtual cache. A Web Proxy server can retrieve objects that were cached by another server in the array.
- Arrays improve fault tolerance since the array continues to function if an individual server fails.

ISA Servers and arrays can be assigned names in DNS. If all servers in an array are given the same FQDN, round-robin addressing helps balance the load between ISA Server clients and the members of the array. In Figure 5, Host Address RRs map the FQDN isaarray1.pseudo to each of the servers in the array. See the section "Supporting Round Robin Addressing" in Chapter 3 for more information.

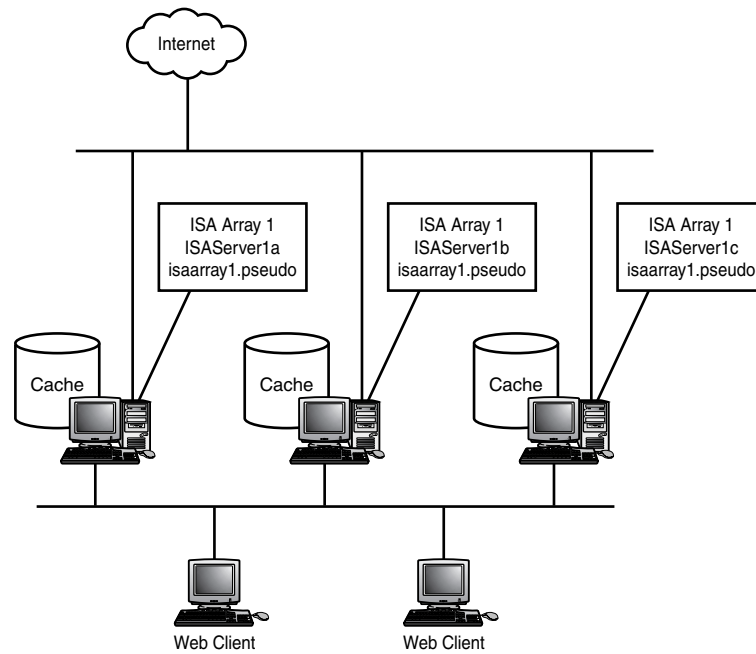


FIGURE 5
An array of ISA Servers.

NOTE

An individual ISA Server can be installed without making it a member of an array. ISA Servers that are not array members are called *standalone ISA Servers*. They do not need to be members of Windows 2000 domains, but if they are not domain members they cannot take advantage of enterprise-level ISA Server features such as enterprise-level policies or Active Directory-based authentication. Standalone ISA Servers are configured using tools available at the array level.

“Enterprise-level” means “within the same Active Directory forest.” All ISA Servers in arrays can share common enterprise-level configuration properties.

Microsoft documentation also refers to ISA Server enterprises and arrays as “scopes,” because they have the effect of determining the extent of the effect for packet filters, rules, and other properties.

In general, it is best to install ISA Servers as Windows 2000 domain members and as array members. A single ISA Server can be installed in a single-server array if it is desirable to give the ISA Server access to Active Directory for authentication.

ISA Server Configurations

ISA Server offers two distinct classes of services:

- Firewall services control the types of packets that are allowed to exit and enter the private network.
- Caching services store objects retrieved from outside servers so they can be supplied to clients locally. They also support the Web Proxy service.

ISA Server can be installed in firewall-only, caching-only, or integrated configurations. The integrated configuration supports both firewall and caching services. Table 1 describes the features that are available in each configuration. These features are described in subsequent sections.

TABLE 1 ISA Server Configurations

	<i>Firewall</i>	<i>Cache</i>	<i>Integrated</i>
Firewall chaining	Yes	No	Yes
Site and content rules	Yes	Yes	Yes
Protocol rules	Yes	Yes (HTTP, FTP, and HTTPS only)	Yes
Web publishing rules	No	Yes	Yes
Server publishing rules	Yes	No	Yes
Routing rules	No	Yes	Yes

Firewall Chaining

In addition to arrays, ISA Servers can also be configured in chains, as in the configuration shown in Figure 6. The ISA Server nearest to the client is known as the *downstream* server. An ISA Server that is closer to the Internet is an *upstream* server. Downstream ISA Servers can pass Internet access requests to upstream ISA Servers, which return the requested data through the downstream servers to the clients.

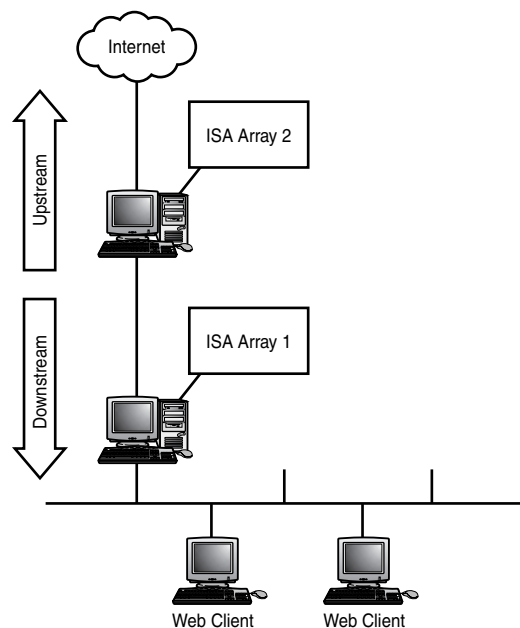


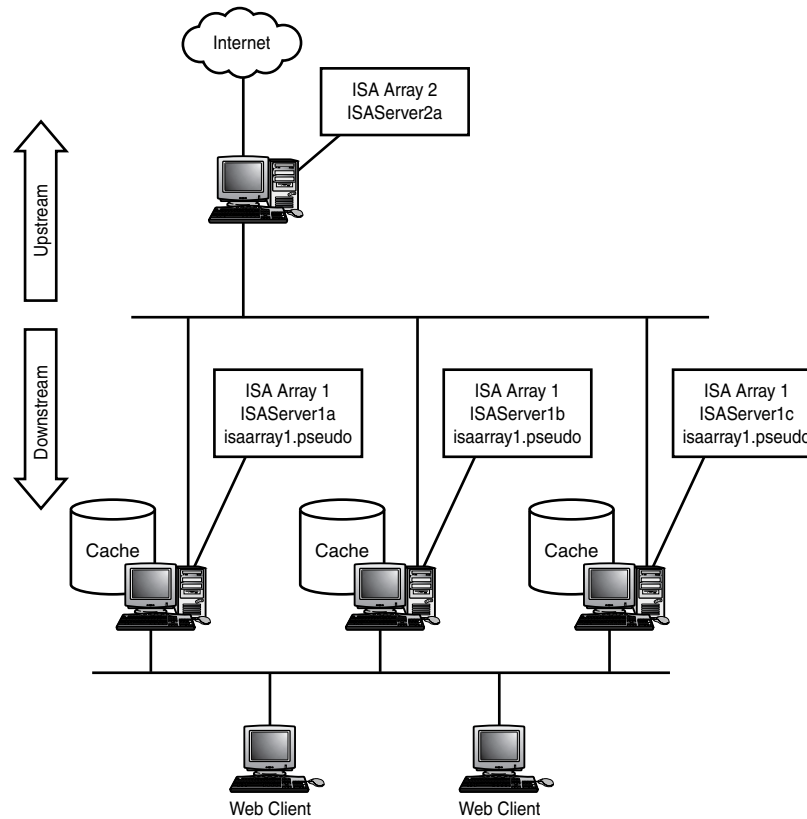
FIGURE 6
Chaining ISA Servers.

Chains of proxy servers provide another means of distributing ISA Server processing. If desired, an ISA Server chain could be combined with a server array, as shown in Figure 7. In this case, ISA Servers at branch offices are chained to an ISA Server array at the home office. The array provides centralized caching and access control for the entire organization while the local ISA Servers cache objects required by local clients. In high-demand environments, chains and arrays can significantly enhance proxy server performance and reliability.

Firewall chaining also enables administrators to configure layers of firewalls that offer varying levels of security. The section “Planning the Network Architecture” illustrates this approach to firewall implementation.

Controlling Access to Services

The security philosophy behind ISA Server is simple: If something isn’t explicitly allowed, it is denied. When first installed, ISA Server does not forward any IP datagrams between the external and internal networks (except for ICMP, DNS, and DHCP). That, of course, makes a connection to the Internet pretty worthless, so ISA Server provides mechanisms that allow specific types of datagrams to be forwarded between the external and internal networks.

**FIGURE 7**

Combining ISA Server chains and arrays.

Forwarding of datagrams is controlled by opening or blocking ports. When the external interface of the ISA Server receives a datagram associated with a blocked port, the datagram is discarded. Two mechanisms determine whether ports are open or blocked:

- Packet filters provide static protocol and port controls that, once enabled, affect every datagram filter criterion. Packet filters can be defined for any IP-based protocol.
- Rules provide dynamic port control, opening and blocking ports based on conditions defined by the administrator. Rules can enable or disable forwarding of outgoing or incoming datagrams based on client identity, permitted protocols, time and day, and so forth. Rules can be defined only for protocols based on TCP and UDP.

Both packet filters and rules can be activated for a given protocol, in which case ISA Server follows specific rules to determine whether datagrams for a particular protocol will be forwarded. We'll look at the decision-making process after seeing what packet filters and rules are.

Packet Filters

Packet filtering can be enabled or disabled. When packet filtering is enabled, packet filters determine which ports will be open and which will be blocked on the ISA Server's external interface. When packet filtering is enabled, the external interface will drop all datagrams that are not explicitly allowed. One way to open ports is to define packet filters that allow the port to be open.

NOTE

Packet filters function at the Internet Protocol layer, and are defined in terms of IP addresses and ports. They do not operate on packets at the network access layer. Because they filter IP datagrams, they would more appropriately be called "datagram filters," but that expression is, so far as I am aware, never used. It's just another instance of inconsistent terminology. It's not my fault. Really!

Packet filtering for a port can have three states:

- **Not defined.** If no packet filter is defined for the port, the port is blocked unless it is dynamically opened by a rule.
- **Block.** If a packet filter blocks access to a port, the port is always blocked, regardless of any packet filters or rules that allow the port to be open.
- **Allow.** If a packet filter allows access to the port, then the port is always open, *provided* no other packet filter blocks access to the port. When a packet filter opens a port, the port is not blocked by rules that deny access.

NOTE

Some objects created in the ISA Server console are RRAS objects and can be viewed in the RRAS console. VPN connections once defined in ISA Server console appear as normal RRAS VPN connections. The same is not true of packet filters. ISA Server packet filters created for ISA Server do not appear as interface packet filters in the RRAS console. To avoid confusion, I suggest that if ISA Server is used on a computer all packet filters be defined in the ISA Server console. Otherwise you have two places to check when resolving filtering problems. Three, actually, because in RRAS each interface can be assigned packet filters.

One advantage of filtering packets with ISA Server is that all defined filters are listed in a table in the ISA Server console rather than being recorded in the properties of individual interfaces. That makes it much easier to manage all the filters on the computer in a cohesive manner. Besides that convenience, ISA Server packet filters have capabilities that aren't available in packet filters defined in RRAS.

A packet filter might specify, “Outgoing DNS queries are allowed from any interface on the ISA server to any server on the external network.”

Another packet filter might specify, “Incoming ping requests are to be dropped.”

Or a packet filter might state that, “Incoming SMTP datagrams are allowed only if they originate on network 203.45.199.0.”

Packet filter definitions include the following properties:

- **Mode.** A filter either blocks or allows forwarding for datagrams matching the filter properties. The mode is declared when the filter is created and cannot be modified afterward.
- **Filter Type.** These properties specify the host-to-host layer protocol (TCP, UDP, ICMP, any, or custom), direction (send, receive, or both), local port(s), and remote port(s) for the filter. The custom filter type enables you to create packet filters for any IP protocol.
- **Local Computer.** This property determines which external interfaces on local ISA Servers are affected by the packet filter.
- **Remote Computer.** This property specifies the IP addresses of one or more remote computers to which the packet filter applies.

By default, packet filters are configured for the following protocols:

- *DHCP client, allow send and receive*, enabling clients to be configured by a DHCP server on either side of the firewall.
- *DNS lookup, allow send and receive*, enabling clients to query DNS servers on either side of the firewall.
- *ICMP ping query, block inbound queries*, preventing outside computers from pinging the firewall or computers on the private network.
- *ICMP (ping request/response, source quench, timeout, unreachable), allow inbound*, enabling internal clients to receive ICMP diagnostic messages generated by outside computers.

The default packet filters allow outbound DHCP and DNS requests to function without being inhibited by the firewall. They also configure the firewall to pass a variety of ICMP messages while blocking incoming ping requests. All other protocols are blocked, however. Given only these packet filters, inside clients wouldn't be able to access any outside servers apart from DHCP and DNS. In most cases, access to services is enabled by defining access policy rules, which control access dynamically based on specified conditions.

NOTE

In Chapter 4, “Active Directory Concepts,” we examined scenarios for deploying DNS with regard to Active Directory. A strong choice is to support Active Directory with a private name space while supporting a public Internet-based name space to facilitate outside access to your servers.

With this scenario, you would probably want to support the private name space on DNS servers that are situated inside the firewall, blocking incoming DNS queries. DNS servers for the public zone would ordinarily be placed outside the firewall to enable outside clients to resolve the organization’s public domain names.

The predefined DNS lookup filter allows DNS queries over UDP, which covers DNS queries made by clients but leaves out other, more complex DNS operations. The `nslookup` diagnostic tool, for example, communicates with DNS over TCP. Zone transfers for standard primary and secondary zones also are made over TCP. A filter to allow DNS over TCP should be defined any time an ISA Server separates clients from their primary DNS servers or DNS servers that engage in zone transfers.

Packet filters can open and block ports to datagrams traveling in either direction through the external interface. This distinguishes packet filters from rules, because a given type of rule affects outgoing or incoming traffic but cannot apply to both.

As already mentioned, packet filters can be defined for any IP-based protocol, that is to say, anything except for ARP (which in any case is not routable). They are most useful for purposes such as these:

- Blocking all use of a particular protocol, even though it may be allowed by a protocol rule.
- Allowing use of a protocol that is denied by a protocol rule.
- Allowing use of protocols not based on TCP or UDP.
- Publishing services when circumstances do not support publishing rules.
- Allowing use of “housekeeping” protocols that must always be forwarded. This is usually the case for DNS, ICMP, and possibly DHCP. Other protocols that are candidates for packet filters include SMTP (if internal mail servers receive mail from the outside), POP3 (if internal users retrieve mail from external mail servers), SNMP (if network management consoles are not on the same side of the firewall as the devices they monitor), and SNTP (if internal devices must communicate with external time servers).

Although packet filters can override rules, it is probably good practice to avoid rules and packet filters that contradict one another. (By contradict, I mean that packet filters and rules that apply in the circumstances but specify opposite actions.) For that matter, it is preferable to avoid rules that contradict other rules and packet filters that contradict other packet filters. Such oppositions make it very difficult to troubleshoot when packets are not forwarded or blocked as desired.

For example, suppose that a protocol rule allows use of HTTP, HTTPS, FTP, and Gopher. To block FTP, we could take one of several actions:

- Remove FTP as a protocol that is authorized by the rule.
- Create a protocol rule that denies FTP under the same circumstances.
- Create packet filters that block ports 20 and 21. (This would block FTP for all clients, not just clients affected by the protocol rule.)

Clearly, the first option is the easiest both to document and to troubleshoot. Rules and packet filters do not clearly announce their properties in the summaries that appear in list boxes, and it is frequently necessary to open up the properties pages to determine exactly what the rule or packet filter does. Consequently, if a protocol rule allows a protocol that is blocked by another protocol rule or by a packet filter, it can be difficult to determine how the block is accomplished. Keep it simple. If you can accomplish a goal with a single rule or packet filter, by all means do so.

Packet Filtering and Routing

Two capabilities interact to determine how packets are forwarded by ISA Server: packet filtering and routing. These capabilities can be enabled separately or together. Each combination has a distinct effect.

Packet filtering determines whether packets are forwarded or dropped by the server's external interface. When enabled alone packet filtering forwards packets using properties assigned to the primary external connection but does not support use of secondary connections.

Routing, (provided as a function of RRAS), supports dynamic route discovery and use of secondary connections. By itself, routing implements no security or protocol control. If routing is enabled without packet filtering, all security on ISA Server is bypassed and the server functions as a non-secured router. To enable ISA Server security it is necessary to also enable support for packet filters.

CAUTION

Microsoft's recommended configuration for ISA Server is to enable both routing and support for packet filters.

NOTE

ISA Server makes use of *routing rules* to determine how client Web requests are routed by the Web Proxy service. Routing rules in the context of ISA Server apply only to the Web Proxy service and do not have anything to do with IP routing as it is performed by RRAS.

Controlling Outgoing Access: Access Policy Rules

ISA Server administrators have considerable control over who does what on the Internet. This control is implemented in the form of *access policy rules* that determine which outgoing requests are allowed and which are denied.

Access policy rules differ from packet filters in that access policy rules take effect only when specific conditions are met. Administrators can define as many access policy rules as are necessary to control outgoing requests to a given service.

When ISA Server is installed, no access rules are activated and all outgoing packets are dropped by the external interface of ISA Server. No outgoing protocols are forwarded until an administrator creates access policy rules that explicitly allow desired protocols to be forwarded.

As with packet filters, access policy rules that deny use of a protocol take precedence over access policy rules that allow use of a protocol.

Access policy rules fall into two categories, which we'll examine in the next two sections:

- *Site and content rules* control access based on conditions such as the identity of the client, the identity of the server, and the time.
- *Protocol rules* determine whether specific protocols are allowed or denied.

NOTE

Packet filters take precedence over rules. Specifically:

- If a packet filter allows the desired service for the desired source and destination, the service request is always allowed, even though it may be denied by one or more rules.
- If any packet filter blocks the desired service for the desired source and destination, the service request is blocked, even though it may be allowed by one or more rules.

Site and Content Rules

Site and content rules determine which Web Proxy clients are allowed to access outside Web servers and the types of objects they can retrieve. Because the SecureNAT and Firewall modules are in most cases configured to forward requests for Web objects to the Web Proxy service, site and content rules affect all users.

Often site and content rules can be very simple. For general access to the Web, for example, a rule might specify, “Anyone can request any type of Web data from any location and at any time.”

But rules can also match narrower conditions. For example, a site and content rule might specify, “Members of the WIDGETS group can request HTTP data from servers on network 192.168.14.0 between the hours of 08:00 and 18:00 on weekdays.”

Alternatively, suppose that you have a group of users who should not be accessing the Web at all. You could define a rule that says, “Members of the SENIOR_EXECUTIVE group are denied access to all types of HTTP content at all times.” (Wouldn’t that be a fun one to enforce?) Keep in mind that these rules affect only requests that are directed to the outside. Users affected by this rule can still access Web servers located inside the firewall.

Site and content rules can have three states:

- **Not defined.** If no site and content rule is defined that matches the conditions of a service request, the request is denied.
- **Denied.** If a service request matches the conditions of any site and content rule that denies access, the service request is denied.
- **Allowed.** If a service request matches the conditions of any site and content rule that allows access, the service request is allowed, *provided* that no applicable rule denies access to the service. Denied trumps allowed every time.

Site and content rules include the following properties:

- **Destination.** This property specifies destinations to which the rule applies. Destinations can be defined in terms of specific IP addresses, network addresses, or sets of addresses defined by an administrator.
- **Schedule.** This property specifies the hours of the day and week during which the rule is applicable.
- **Action.** This property determines whether the rule allows or denies access if the conditions in the rule are met.

- **Applies To.** These properties define the clients to which the rule applies. A rule can apply to all clients, to specific sets of IP addresses, or to specific users or groups.
- **HTTP Content.** This property specifies the types of HTTP content data to which the rule applies. HTML documents, audio, video, and images are examples of HTML content data types.

Destination, Schedule, and Applies To properties clearly distinguish access control rules from packet filters. The only conditions available for packet filters are source and destination addresses, and the filter defined for these addresses applies 24x7, regardless of who is using the computer. Site and content rules (and other rules as well) can be defined to apply only at certain times or only to certain users or groups, for example, so that a user connecting from computer 10.1.1.205 during business hours might be governed by different restrictions than the same user connecting from the same computer on weekends.

Protocol Rules

Protocol rules function like site and content rules but determine whether and when access is allowed or denied for specific network protocols. If you don't want certain users to be using AOL Instant Messenger, for example, you can block the protocol with a protocol rule. Protocol rules apply to SecureNAT and Firewall clients only. (SecureNAT and Firewall clients are also affected by site and content rules for requests that are directed to the Web Proxy service.)

A protocol rule can have three states:

- **Not defined.** If no protocol rule is defined for a service, ISA will drop any requests for the service. Unless a protocol rule explicitly allows access to the service, access is denied.
- **Denied.** If any rule explicitly denies access to a service, ISA will drop service requests from the client. However, if any packet filter allows use of the port, the port is always open and cannot be blocked by a rule that denies access.
- **Allowed.** If any rule allows access to the port, then ISA Server will forward outgoing datagrams for the client via the port, *provided* no other protocol rule denies the client access to the port. Also, if any packet filter blocks the port, ISA Server will never forward datagrams for that port.

Protocol rules include the following types of properties:

- **Action.** This property determines whether the rule allows or denies access if the conditions in the rule are met.
- **Protocol.** These properties define the protocols that are matched by the rule.

- **Schedule.** This property specifies the hours of the day and days of the week during which the rule is applicable.
- **Applies To.** These properties define the clients to which the rule applies. A rule can apply to all clients, to specific sets of IP addresses, or to specific users and groups.

NOTE

Protocol rules can be created only for protocols based on TCP or UDP. Access to other protocols, including ICMP, must be controlled using packet filters.

Access Policies

Access policies are simply collections of rules that simplify the process of associating rules with ISA Servers. If you want to configure several ISA Servers in the same way, you simply assign the same access policy to each server. Access policies can be defined at two levels:

- *Enterprise access policies* apply to none, some, or all of the ISA Server arrays in an Active Directory forest, depending on properties established at the enterprise and array levels. Enterprise access policies incorporate the following access controls:
 - Site and content rules
 - Protocol rules
- *Array access policies* are assigned to a single array. Array access policies include the following access controls:
 - Site and content rules
 - Protocol rules
 - Packet filters

Note that packet filters can be defined only for arrays or for standalone ISA Servers (which are configured as single-member arrays). They cannot be defined for multiple arrays via enterprise access policies. This is the case because packet filters may refer to the IP address of a specific ISA Server and for that reason cannot be assumed to be universally applicable.

ISA Server can be configured to support access policies in three modes:

- *Enterprise access policies only.* Properties established in the enterprise access policy cannot be overridden by properties in array access policies. Array access policies have no effect. This approach centralizes management of ISA Server arrays.

- *Array access policies only.* Each array is configured by an individual access policy. Properties established at the enterprise level have no effect. This approach distributes ISA Server management to administrators of individual arrays.
- *Both enterprise and array access policies.* Default access properties can be defined in enterprise access policies. Array access policies may override. This approach enables central administrators to define default access policy properties that can be customized by administrators of arrays. Array policies cannot be less restrictive than the policy established for the enterprise containing the array.

Policy Elements

Policy elements are pre-defined groups of parameters that can be added to some rule properties. For example, a policy element can be defined that contains the IP addresses of all of the ISA client computers in the Marketing department. This policy element can then be added to rule properties that accept client address specifications, perhaps to limit access for marketing plans to members of the Marketing group.

Policy elements may be defined at two levels:

- *Enterprise-level policy elements* are shared throughout the enterprise and may be used in rules in any enterprise or array access policy. Even though multiple enterprise policies can be defined, there is only one set of enterprise-level policy elements, which is shared globally by all enterprise and array access policies in the enterprise.
- *Array-level policy elements* can be used only with rules in a particular array. Some types of policy elements can only be defined at the array level.

Seven types of policy elements can be defined:

- **Bandwidth Priorities.** Bandwidth priorities determine scheduling priorities for connections. For example, email connections, which typically are not real-time, might be given lower-than-normal priority, whereas streaming audio or video might be given higher-than-normal priority, that is, most ready access to bandwidth. (Array level only)
- **Client Address Sets.** Client address sets are groups of client IP addresses that enable access policies to affect multiple clients. (Enterprise and array levels)
- **Content Groups.** Content groups define types of content delivered via HTTP. Content groups are used in site and content rules. (Enterprise and array levels)
- **Destination Sets.** Destination sets are groups of server names or IP addresses that enable access policies to affect connections to several servers. (Enterprise and array levels)
- **Dial-Up Entries.** Dial-up entries group dial-up connections (as defined in the Network and Dial-Up Connections applet) with the associated user name and password. Dial-up entries are used in definitions of VPN connections. (Array level only)

- **Protocol Definitions.** Protocol definitions describe protocols that can be allowed or denied in protocol rules. Protocol definitions can be created only for protocols based on TCP or UDP. Protocol definitions are used in protocol rules and server publishing rules. (Enterprise and array levels)
- **Schedules.** Schedules determine when a connection can be used by hour of the day and day of the week. (Enterprise and array levels)

Policy elements have no effects by themselves. They are simply pre-defined data sets that can be entered in properties of rules and filters. In most cases, policy elements are required when it is necessary to assign more than one value to a property.

Summary of Outgoing Access Controls

Access to outside services is controlled by three access control components, which ISA applies in the following order:

1. *Protocol rules* affect SecureNAT and firewall clients.
2. *Site and content rules* affect Web Proxy clients and requests sent to the Web Proxy service by the SecureNAT and firewall services.
3. *Packet filters* affect all clients.

Each of these controls can have three states, which ISA Server considers in the following order:

1. Blocking or denying
2. Not defined
3. Allowing

ISA Server uses the following logic to determine whether a client is given access to an outside service:

1. For SecureNAT and firewall clients, ISA Server evaluates the protocol rules. Requests that cannot be directed to the Web Proxy service are denied if either of the following conditions exists:
 - Any protocol rule denies the request
 - A protocol rule that allows the request is not found
2. For Web Proxy clients, ISA Server evaluates the site and content rules. The request is denied if either of the following conditions exists:
 - Any site and content rule denies the request
 - A site and content rule that allows the request is not found
3. ISA Server evaluates the packet filters. If any packet filter blocks the requested protocol the request is denied.

If the request is allowed, it can be forwarded only if ISA Server identifies a route to the target server. In the majority of cases, outside servers will be on the Internet and the ISA Server's default route is sufficient to enable the request to be delivered.

In some cases routing rules must be defined to enable the Web Proxy service to successfully direct requests to the desired Web server.

Figure 8 summarizes the process by which ISA Server determines whether an outgoing connection request should be allowed and the results of using different access control combinations.

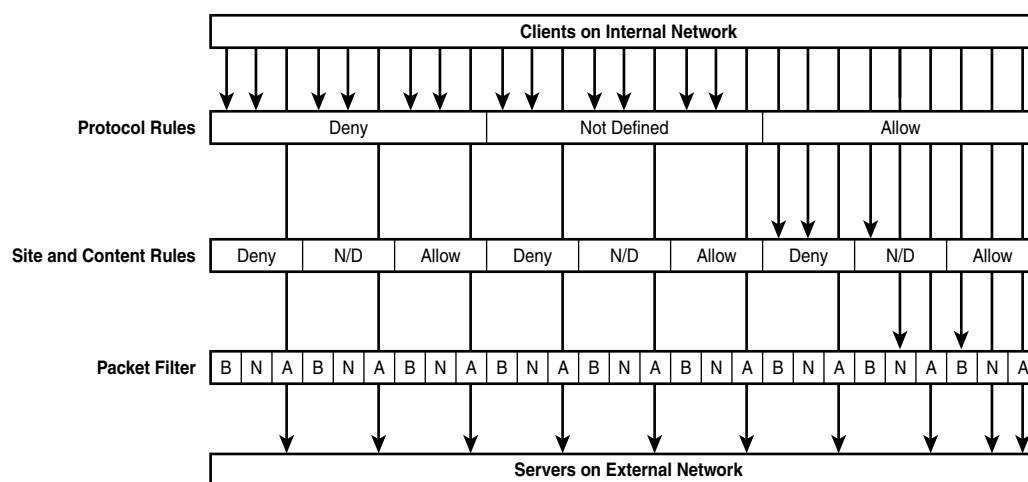


FIGURE 8

Access protocol rules and protocol filters both affect outgoing service access.

Controlling Incoming Requests: Publishing Policy Rules

The primary reason for implementing a firewall such as ISA is to prevent outsiders from improperly accessing resources on a private network. By default, ISA does not permit outside computers to direct service requests to any local computers. But suppose we have a Web server that we want to make available for public access. We don't want it wide open since that would make it too easy to attack. We need a way to put the Web server behind the firewall while allowing outside users to access it in carefully controlled ways.

Services running on inside servers are made available to outside users by *publishing* them in ISA Server. Publishing is accomplished by establishing *publishing policy rules* that specifically identify services to be published and how and by whom the services may be accessed.

Publishing policy rules fall into two categories:

- *Web publishing rules* apply only to service requests directed to Web servers.
- *Server publishing rules* can apply to any type of server, but are usually employed to publish non-Web services.

Web Publishing Rules

Outside clients direct all requests for Web services at your site to ISA Server. Two components enable ISA Server to publish internal Web servers and make them available to outside clients:

- *Publishing rules* enable ISA Server to accept these Web requests on behalf of a local Web server. ISA Server becomes the public access point for Web servers on the private network.
- *Routing rules* enable ISA Server to direct Web requests it receives to the appropriate Web server. These rules also determine whether port numbers need to be remapped when datagrams are forwarded.

NOTE

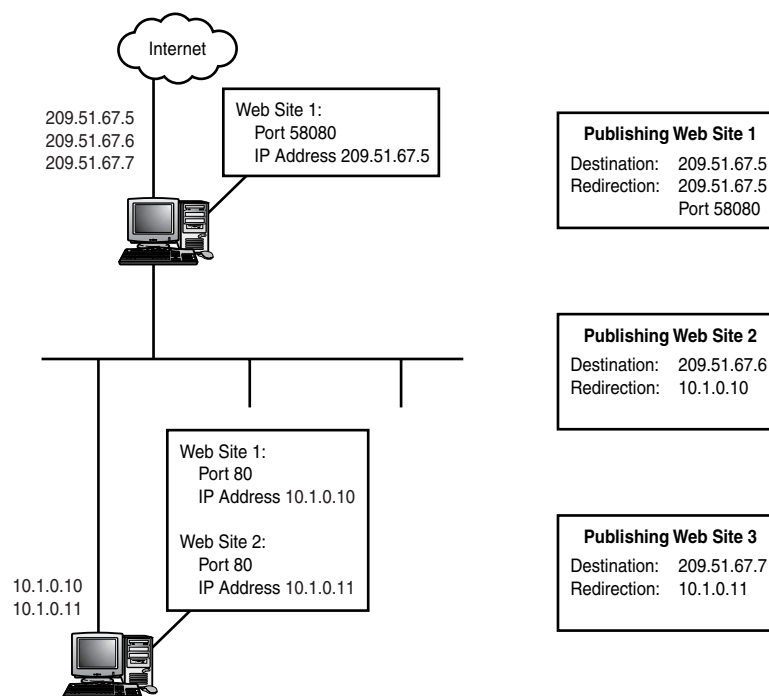
Reverse caching is automatically enabled to support incoming Web requests.

In many cases, organizations will have several Web servers that are published to outside users. These Web servers are differentiated by their IP addresses, port numbers, or both. Figure 9 shows a network with three private Web servers. One is running on the same computer as the ISA Server. Two, running on a separate, dedicated Web server are differentiated by separate IP addresses.

To enable outside users to access these Web servers, ISA Server offers three IP addresses, each of which maps to a particular Web server. Web publishing rules configure ISA Server to receive Web requests on these addresses. Routing rules enable ISA Server to redirect Web requests to the correct server.

The first step in such situations usually is to logically multihome the ISA Server's external interface with the Internet by assigning an IP address for each Web server to the interface. (In interface **Properties**, use the **IP Setting** tab in the **Advanced TCP/IP Settings** dialog box.) After a distinct DNS name is assigned to each IP address, as far as users are concerned each IP address identifies a distinct Web server.

After interfaces are configured, Web publishing rules and routing rules can be defined.

**FIGURE 9**

Supporting multiple Web servers with Web publishing rules.

In some cases, the physical Web server may not be configured to use the standard ports for HTTP, SSL, or FTP. When that happens, the Web publishing rule can be configured to map the port number used by outside clients to the port number actually configured on the Web server. This must be done in the case of any Web server running on the computer with the ISA Server. Port 80 is already used to accept HTTP requests from the outside, so the Web server must be assigned a different port number. In the example of Figure 9, I used port 58080, which is selected from the range of ports that is designated for private use. The Web publishing rule that is defined for this Web server specifies the port translations that are to take place when requests and responses are forwarded between the client and the Web server.

Web publishing rules include these properties:

- **Destinations.** This property specifies the destination of the incoming Web request, which is usually the IP address on the public network that is used to access the Web server.
- **Actions.** These properties specify redirection rules for the incoming Web request. The destination Web server is identified by its IP address along with any required port mappings.

- **Redirection.** It is possible to communicate between the ISA Server and the internal Web server using a different protocol than the client used to submit the request. For example, secure HTTP can be used to communicate between ISA and the Web server even though the incoming request was unencrypted HTTP. Alternatively, ISA Server can require outside clients to communicate via HTTPS while communicating with the internal server using HTTP. These properties specify when SSL communications are required between the ISA and Web servers and how certificates are used in the connection.
- **Applies To.** These properties define the clients to which the rule applies. A rule can apply to all clients, to specific sets of IP addresses, or to specific users and groups.

ISA Server evaluates Web publishing rules according to the order specified by the administrator. A default Web publishing rule is included in the ISA Server configuration. This rule cannot be modified and is always positioned as the last rule in the Web publishing rule list. It instructs the ISA Server to discard all incoming Web requests. If no Web publishing rules are defined by the administrator, the default rule ensures that all Web requests are discarded.

The use of SSL when communicating with the ISA Server requires additional discussion. We'll pick up the topic again in the section "SSL Tunneling and Bridging."

NOTE

ISA documentation refers only to SSL, not to TLS, so I'm using that term in this chapter. As explained in Chapter 11, Windows 2000 supports TLS as well as SSL, and any remarks made in this chapter apply to both protocols.

Server Publishing Rules

Web servers aren't the only servers that can be published by the ISA Server. Server publishing rules enable a wide variety of services to be published. ISA Server supports publishing for services in the following list, which presumably can be extended by vendors who want their products to be supported by ISA Server:

- Exchange RPC (Microsoft Exchange Remote Procedure Call)
- Any RPC (Windows's native messaging protocol)
- FTP
- RTSP (Real Time Streaming Protocol, used by Real Player G2 and QuickTime 4)
- PNM (Progressive Networks Protocol, used by RealNetworks)
- MMS (Microsoft Windows Media, used by Windows Media Player)
- SMTP

- DNS (Query and Transfer Zone)
- HTTPS
- IMAP4
- NNTP
- POP3
- Telnet

A server publishing rule includes these properties:

- **Action.** These properties specify the public IP address that is used to access the service, the IP address of the server providing the service, and the identity of the service.
- **Applies To.** These properties define the clients to which the rule applies. A rule can apply to all clients, to specific sets of IP addresses, or to specific users and groups.

The only conditions that can be defined in a server publishing rule are the addresses of clients that are permitted access by the rule. Server publishing rules cannot authenticate clients. Authentication, if desired, must be performed by the target server.

Server publishing rules cannot be configured to block or allow a protocol. If the rule exists, it is an allow rule. To disable external access to the service you can

- Disable the server publishing rule
- Delete the server publishing rule
- Define a packet filter that blocks the port used by the server publishing rule

In some cases, protocol filters must be configured to publish internal services.

Summary of Incoming Access Controls

Access to internal services by outside clients is controlled by three access control components, which ISA applies in the following order:

1. Packet filters
2. Web publishing rules or server publishing rules
3. Routing rules

Each of these controls can have three states, which ISA Server considers in the following order:

1. Blocking or denying
2. Not defined
3. Allowing

ISA Server uses the following logic to determine whether an outside client is given access to an inside service:

1. ISA Server evaluates the packet filters. If any packet filter blocks the requested protocol, the request is denied.
If any packet filter allows the requested protocol, the request is allowed without evaluation of Web publishing rules or server publishing rules.
2. ISA Server evaluates the Web publishing rules or the server publishing rules as appropriate. The request is denied or allowed as follows:
 - If any Web or server publishing rule denies the request, the request is denied
 - If no applicable Web or server publishing rule exists, the request is denied
 - If a Web or server publishing rule allows the request, the request is allowed

If the request is allowed, it can be forwarded only if ISA Server identifies a route to the target server. On internal networks, routes must usually be defined for each local subnet, because the default router option is used to direct traffic to the Internet.

Figure 10 summarizes the process by which ISA Server determines whether an inbound connection request should be allowed and the results of using different access control combinations.

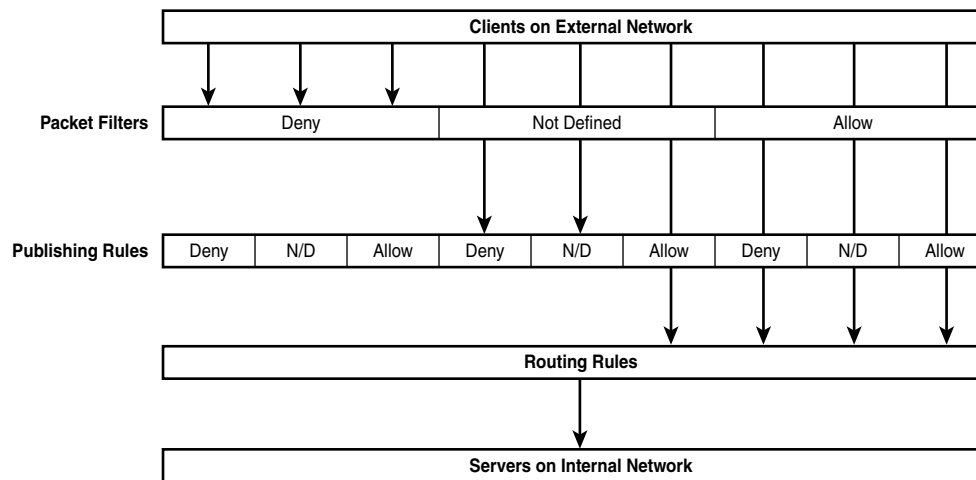


FIGURE 10

Access protocol rules and protocol filters both affect inbound service access.

Routing Rules

Routing rules, not to be confused with IP routing as maintained by RRAS, determine how the Web Proxy service forwards Web requests from internal clients to external Web servers.

Routing rules can be defined to do the following:

- Send client requests directly to the destination specified in the request
- Send client requests to an upstream server in a server chain
- Redirect client requests to an alternative Web site

Routing rules can be used to configure alternative routes so that clients are serviced if a preferred route fails.

In addition, routing rules can instruct ISA Server to communicate with the client or the Web server using SSL. One use of this capability enables ISA Server to secure outside Web communications even if the client's request does not specify use of HTTPS. The section "SSL Tunneling and Bridging" provides more information on the use of routing rules.

Bandwidth Rules

WAN links are often the conduits for a wide variety of network traffic. On many occasions, administrators would like control over the manner in which that bandwidth is allocated so that high-priority processes do not have their bandwidth preempted by processes that don't have to take place in real time, such as routing of email.

Bandwidth rules work with the Windows 2000 Server Quality of Service (QoS) packet scheduling service to enable administrators to set limits on bandwidth that is available to specific processes. This can be used to guarantee bandwidth for high-priority processes such as IP telephony or video conferencing. It can also be used to ensure that data-intensive but low-priority tasks such as file transfer don't eat up too much bandwidth. It matters little if a file takes a bit longer to download, but it matters a great deal if streaming video is disrupted.

Intrusion Detection

One thing we like a firewall to do is warn us when an apparent break-in takes place. Frankly, if you're connected full time to the Internet, you're going to be the target of some probes that seek to locate systems with security holes that can be easily exploited. Properly configured, ISA Server will block most routine intrusion attempts, but it is still nice to see a red flag when someone is trying to gain access.

Intrusion detection is a function of ISA Server's routing capability. ISA Server can detect and report the following types of intrusion events:

- All ports scan probes
- IP half scan attacks
- Land attacks
- Ping of death attacks
- UDP bomb attacks
- Windows out of band attacks

These attacks are ridiculously easy to perform. They exploit simple holes in the TCP/IP protocols or in protocol stack implementations. Browsing hacker Web sites will turn up source code and even compiled programs that reduce the attacks to elementary procedures that any dumb, malicious jerk who can push a mouse can find the brain cells to perform. So these attacks aren't cutting edge. In fact, they're rather old news as far as serious hacking goes.

In fact, Microsoft has released patches and Service Packs that plug the loopholes that allow many of these hacks to disrupt Windows NT 4 and Windows 2000. But just because they aren't a problem doesn't mean you don't want to be aware of the attacks. Some intruders will throw a battery of attacks at a target site trying to identify exploitable weaknesses. Frequent occurrences of these attacks can function as canaries in a coal mine, letting you know that something nasty is seeping in that could kill you if you ignore it.

Several of these attacks can be difficult to detect because they cause vulnerable computers to crash and are, therefore, not logged. It can be difficult even to tell whether a network-based attack was the cause of the system failure. Intrusion detection can provide valuable evidence that an attack has taken place and may assist with blocking intrusion efforts and possibly with identifying the perpetrator.

Let's examine each of these events so that you understand what is happening when ISA Server sounds an alarm.

All Ports Scan Probes

Someone who is trying to break into your network may try to poll all or a large number of ports, to see which ports respond. This can provide clues regarding the type of computer and the services that are operating on it. For example, port 139 is assigned to NetBIOS. Port 1512 is assigned to WINS. A computer with these ports open almost has to be a Windows server. A sophisticated hacker might try some Windows-specific techniques on such a computer. If you search for "Microsoft" in the ports list on www.iana.org, you'll see a number of ports that are registered to Microsoft services.

But port knowledge can aid a cruder form of attack. In a SYN flood attack, the intruder barges the server with large numbers of SYN packets, opening connections that hog server resources. To work, the attack must target an open port. These attacks can be harder to detect and filter out if the SYN packets are directed at many different ports rather than a single port.

ISA Server reports a port scan probe if probes are attempted on more than a specified number of ports. The threshold can be configured by the administrator to adjust the sensitivity.

NOTE

To reduce vulnerability, a computer connected to the Internet should not have any open ports that are not needed for required functionality. This can be tested by running a port scan of your own.

One way to test a computer is to connect to <http://grc.com> and run their ShieldsUp diagnostic. ShieldsUp will scan your ports and report all ports that are open. This is, incidentally, the site for Steve Gibson, legendary creator of SpinRite. Steve is taking Internet security very seriously and is currently developing OptOut, a program that examines a computer for the presence of spyware. The term spyware refers to programs that are installed on a computer from a Web site, often without the user's knowledge, that periodically report user activities to a site, again without the user's knowledge. Some spyware is innocuous, while some is really invasive. Spyware may become a serious problem on corporate networks, both from a security and a network bandwidth standpoint. Take the time to browse Steve's site and learn about his products.

A wonderful commercial port scanner is available from www.atelierweb.com. The AW Security Port Scanner can be downloaded for a fifteen-day trial. The single-user license is \$35, and a company license is only \$80. Better bargains are hard to find.

IP Half Scan Probes

A half scan attack involves a violation of the SYN/SYN-ACK/ACK message sequence that is used to set up a TCP connection. (See "Connection-Oriented Delivery" in Chapter 2 for a discussion of the connection setup procedure.) The client sends a SYN message to the server. The server responds either with a SYN-ACK, indicating that the port is open to accept connections or with a RST (Reset), which indicates that the port is not open for connections. Consequently, the intruder learns which ports are open on the target computer.

The advantage of this form of attack is that many systems do not log connections until they receive the final ACK message. Consequently, the SYN/SYN-ACK and SYN/RST exchanges aren't logged and the intruder doesn't raise any flags. If the attack is undetected, the intruder gets away with a list of ports with no one the wiser.

ISA Server will detect these probes and generate an alert, enabling you to identify the source of the attack. The attacker can be blocked with a packet filter and, because the intruder must receive a reply and therefore must include its real IP address in the SYN packet, it is often possible to identify the attacking computer and have the user shut down by contacting the user's ISP.

Land Attacks

With this attack, the intruder sends a SYN message with an IP source address and port that are the same as the destination IP address and port. On some older systems, this type of packet would cause the computer to attempt to open a connection with itself, which could crash the system.

This type of packet is never valid and should never be accepted by the protocol stack. ISA Server could be configured with an incoming packet filter that blocks packets with the server's own IP address in the source IP address field. However, the land attack is a rather old trick and Windows has been immune to it for quite some time. Recent Windows versions are not vulnerable. Windows 98, Windows 2000, and Windows Me are resistant out of the box.

To fix the problem on Windows 95, install WinSock version 2, available at www.microsoft.com. (See Knowledge Base article Q177539.) The problem does not affect Windows 95 OEM release OSR2. To correct the situation on Windows NT 4.0, install Service Pack 3 or later. (Service Pack 3 introduced a new TCP/IP protocol stack for Windows NT 4.0.)

Ping of Death Attacks

The ping of death attack involves sending an ICMP echo request (ping) packet that is oversized and must be fragmented on the network. A vulnerable computer that receives the fragments and attempts reassembly may experience a buffer overflow. The results are a bit unpredictable. The computer may stop operating or may reboot.

This attack has not been a particular problem for Windows. Windows 3.51 requires Service Pack 4 or later to deal with it. Windows 95/98, NT 4.0, and 2000 are not vulnerable.

UDP Bomb Attacks

A UDP bomb attack involves a UDP packet that has illegal values in certain UDP header fields. Affected systems may crash when they receive the packet.

Windows Out of Band Attacks

An out of band attack consists of a packet in which the Urgent Pointer field in the TCP header is configured illegally. It is most often called *Winnuke*, because it specifically targets vulnerabilities in Windows 95. When the Urgent flag is set (equals 1), the Urgent Pointer field is significant. The Urgent Pointer field identifies the sequence number of the octet following urgent

data. Packets with invalid Urgent Pointer fields can crash Windows 95 computers. This attack often arrives on port 139, which is assigned to NetBIOS.

See Microsoft Knowledge Base article Q168747 for corrective procedures.

NOTE

The attacks detected by ISA Server aren't terribly significant threats to current Windows versions. More threatening attacks appear all the time, and you should periodically consult www.microsoft.com/security/ to see if Microsoft has issued any security alerts or patches relevant to your organization. If enabled, the Windows Update capability included in Windows 2000 will occasionally notify you of patches Microsoft has issued to correct newly discovered vulnerabilities.

Denial of Service Attacks

ISA Server offers little help with the most damaging class of attacks, grouped under the heading Denial of Service (DoS), because these attacks take advantage of vulnerabilities in the fundamental operations of TCP/IP protocols.

For example, servers must accept SYN packets or connections cannot be opened. The SYN flood is an elementary attack that bombards the server with SYN packets. The server starts a connection dialog for each SYN packet and allocates connection resources that remain tied up until the connection attempt times out. Eventually, resources become short and are unavailable to process legitimate connection requests. SYN attacks are detected when a firewall notes large numbers of SYN packets arriving from the same source. The source IP address is usually spoofed, making it difficult to identify the perpetrator, but filters can be enabled to block packets from that source.

A newer, more devastating form of the SYN flood attack is the *Smurf* attack. This attack elicits the assistance of a third-party network, making an innocent organization's network the source of the attack.

A Smurf attack begins by sending an ICMP echo request to a directed broadcast address. These addresses take the form *n.n.255.255*, which means "broadcast this datagram to all devices on network *n.n.0.0*." The source IP address in the ICMP echo request packet is spoofed with the IP address of the victim computer. When the attack is initiated, every active device on the intermediate network responds by sending an ICMP echo response to the victim computer. This technique amplifies the single ICMP echo request packet by the number of computers on the intermediate network.

In very intense Smurf attacks, several intermediate networks can be involved, generating enough packets to overwhelm any server. This technique was most notably used to target Internet online retailers during the 1999 Christmas season. The similar Fraggle attack causes similar disruptions using UDP packets.

Because the perpetrator spoofs the address in the original packet, and the packet floods originate from innocent intermediaries, it can be very difficult to shut down a Smurf attack. Filters can be added to incoming firewalls, but the numbers of packets can bog down even the routers that are configured to filter them out.

At some point, network owners may be held liable if they have not configured their networks so that they cannot become involved in Smurf and other such attacks. The general approach is to configure border routers to not forward directed broadcast packets. RFC2644 contains IETF recommendations for configuring routers to reduce the incidence of DoS attacks.

Unfortunately, ISA Server does not provide filtering capability for directed broadcast messages. That won't matter for small organizations, which are the primary market for ISA Server, and it may not matter if only a few of your computers are directly exposed to the Internet. Large organizations with many exposed computers will want to resort to a hardware firewall solution that supports RFC2644 recommendations.

Client Authentication

ISA Server supports several methods of authenticating clients. Authentication is a property that can be included in rules to determine whether the client is permitted to request a particular service from a particular server. Authentication distinguishes SecureNAT from plain old NAT, which cannot block or allow packet forwarding based on the identity of the client.

The ISA Server can share authentication with the target Web server using pass-through authentication. It can also forward authentication information when a request is sent to an upstream ISA Server using chained authentication. After examining the authentication protocols supported by ISA Server, we'll see how pass-through and chained authentication are used.

Authentication Protocols

ISA Server supports four modes of client authentication:

- Basic authentication
- Digest authentication
- Integrated Windows authentication
- Client and server certificates

While supporting secure authentication, these methods enable ISA Server to authenticate Windows and non-Windows clients at a variety of security levels. Internet Explorer versions 5 and higher support all of these authentication methods.

Basic Authentication

This is the standard authentication method used, for example, by unsecured Web and FTP servers. When authentication is required the client prompts the user to enter a user name and password. The Web Proxy server verifies the user credentials with Active Directory if it is a member of a Windows 2000 domain or, if it is a standalone server, with its local user accounts. Consequently, basic authentication can be used with ISA Servers that are not members of Windows 2000 domains.

Authentication information is encoded for transmission to the Web Proxy server, but no encryption is used. While on the wire, this information is protected from casual examination but cannot be regarded as secure. Although it is not secure, basic authentication is supported by all Web clients and may be the only authentication method that can be used in some cases.

Digest Authentication

Digest authentication functions similarly to basic authentication except that authenticators are processed by a hashing algorithm before transmission. As explained in Chapter 9, "Data Communication Security Concepts," message digests generated by secure hashing algorithms are very well protected, and it is virtually impossible to recover the original authenticators from the message digest. Digest authentication is supported only within Windows 2000 domains.

The message digests used with digest authentication include values that identify the computer that originates the request and its domain as well as a time stamp. This information prevents replay of the packet at a later time.

Integrated Windows Authentication

Integrated Windows Authentication can use Kerberos v5 or MS-CHAP to authenticate users. Both methods encrypt authenticators on the wire. The client and ISA Server must be a member of a Windows 2000 domain to use this authentication method.

Client and Server Certificates

This method uses certificates and SSL to authenticate both the client and the ISA Server. The client and server must each have a certificate issued by a trusted certification authority. See Chapter 10, "Planning and Implementing a Public Key Infrastructure," for information about issuing and installing certificates. See Chapter 11, "Securing IP Communication," for discussion of the SSL and TLS protocols.

Pass-Through Authentication

Pass-through authentication occurs when ISA Server authenticates a user and passes the credentials to the server with which the client is attempting to connect. The ISA Server may initiate authentication if a rule requires it. However, the target server may also respond to the client request by requesting client authentication. In its role as intermediary, the ISA Server obtains credentials from the client and forwards them to the server. That process is illustrated in Figure 11.

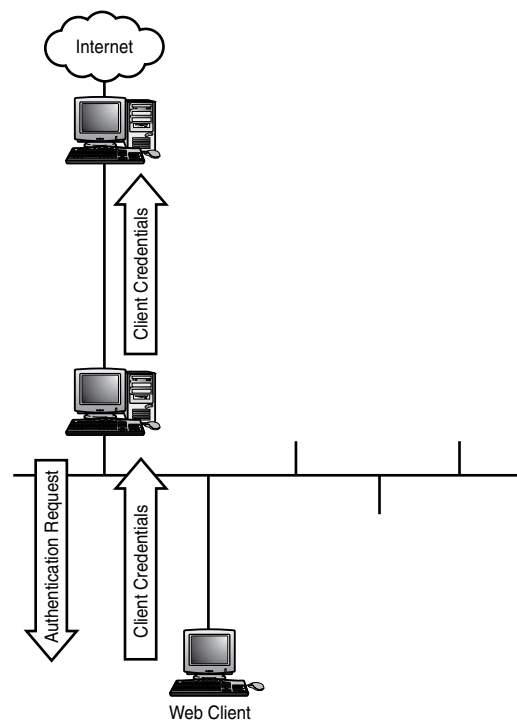


FIGURE 11

Pass-through authentication allows ISA Server to obtain credentials required by a server.

Because Kerberos v5 requires that the client authenticate the server, it is not supported as an authentication method with pass-through authentication. This will be an issue primarily with IIS, which can use Kerberos v5 as an authentication protocol. Some other authentication method will be required. Digest authentication, MS-CHAP, or certificates could be employed in these circumstances.

Chained Authentication

When ISA Servers are configured in chains, an ISA Server can pass authentication to an upstream ISA Server, a process called *chained authentication*. Chained authentication is supported for Proxy Server 2.0 and ISA Server upstream servers.

In some cases, the upstream ISA Server may not be able to authenticate the user from the credentials provided by the downstream server. This might be the case if the upstream server is configured in stand-alone mode and therefore has a different account database than the downstream server. In such instances, the downstream ISA server can send its own credentials. These credentials are defined by an administrator as static properties of the downstream ISA Server.

SSL Tunneling and Bridging

ISA Server supports use of SSL to secure communication among the client, ISA Server, and the external Web server. This support can take several forms:

- The client requests an HTTPS connection with a secure Web server.
- The client communicates with ISA Server using HTTP. ISA Server communicates with the Web server using HTTPS.
- The client communicates with ISA Server using HTTPS. ISA Server communicates with the Web server using HTTP.
- The client communicates with ISA Server using HTTPS. ISA Server establishes a separate HTTPS connection with the secure Web server.

To cover these scenarios, ISA Server supports two techniques: SSL tunneling and SSL bridging.

NOTE

The documentation for ISA Server uses the acronym SSL to describe the method of securing HTTP communication. Here, because the technique comprehends both the Secure Sockets Layer and the Transport Layer Security protocols, I'm using the expression SSL, following the convention I used in Chapter 11, which in turn follows conventions established in the *Windows 2000 Server Resource Kit*. If you're searching for information in ISA Server Help, search for SSL.

SSL Tunneling

When a Web client that communicates with ISA Server via unsecured HTTP requests a connection with an external secure Web server, ISA Server establishes a tunnel to support

communication between the client and the external Web server. Figure 12 illustrates an SSL tunnel. The tunnel setup takes place as follows:

1. The client sends an HTTPS service request to port 8080 (by default) on the ISA Server; for example, the client sends the request to `https://www.pseudo-corp.com`.
2. On behalf of the client, the Web Proxy sends the request to port 443 on the target server.
3. A connection is negotiated with the secure Web server.
4. When the connection is established, the ISA Server sends a connection established message to the client.
5. An SSL tunnel is now established between the client and the target Web server. All further communication takes place directly between the client and the Web server via the tunnel.

While the SSL tunnel persists between the client and server, ISA Server is effectively bypassed. This should improve efficiency since ISA Server is not required to edit packets or to maintain connections with the client and server while data are exchanged.

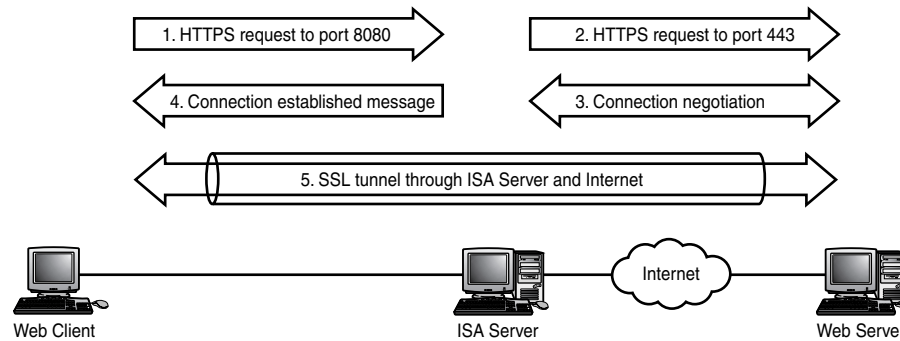


FIGURE 12

Once established, an SSL tunnel enables the client and server to communicate directly, bypassing ISA Server.

ISA Server does not build SSL tunnels to support incoming HTTPS requests. For incoming requests, ISA Server always resorts to bridging, which we take up next.

NOTE

Tunneling is the default method used to support outgoing client requests to ports 443 (HTTPS) and 563 (secure Network News Transfer Protocol, [NNTPS]). Additional SSL ports can be defined by setting the ISA Admin COM object `FPCProxyTunnelPortRange` using procedures described in documentation for the ISA Server SDK that is included with ISA Server.

SSL Bridging

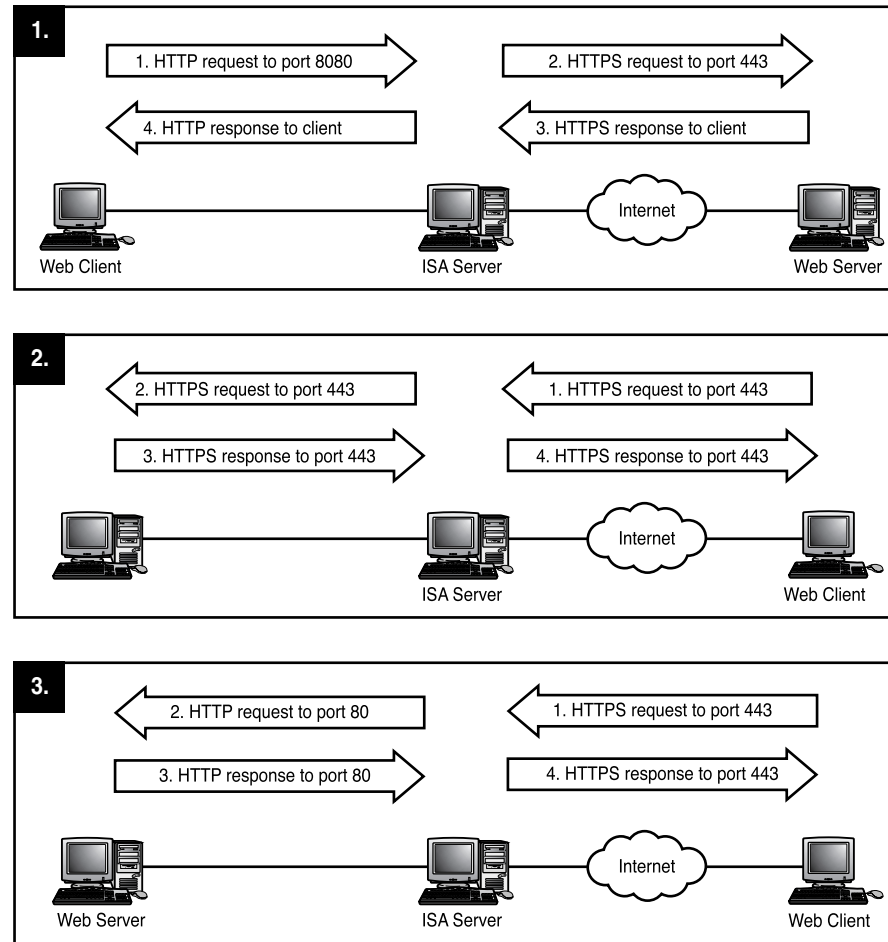
When an SSL tunnel cannot be established, ISA Server resorts to SSL bridging to facilitate client-server communication. Bridging requires more processing by ISA Server because it must encrypt or decrypt most or all of the packets involved in client-server communication. However, SSL bridging enables ISA Server to support SSL security in situations when it would not normally be invoked.

SSL bridging is employed any time ISA Server ends or initiates an SSL connection. Bridging is employed in three situations, illustrated in Figure 13:

1. A Web Proxy client sends an unsecured HTTP request to port 8080 on the ISA Server. The ISA Server encrypts the request and sends it to port 443 on the secure Web server, which returns a secure response. ISA Server receives the encrypted response from the Web server, decrypts it, and returns the unencrypted response to the client. This scenario permits the administrator to require SSL communication between ISA Server and the Web server so that all communications with the Web server are encrypted on the public network.
2. An external client sends a secured HTTP request to port 443 on the ISA Server. The ISA Server decrypts the request. It then encrypts the request and sends it to port 443 on an internal secure Web server. ISA Server receives the encrypted response from the Web server and decrypts it. Then ISA Server encrypts the response, and returns the encrypted response to the client.
3. The client sends a secured HTTP request to port 443 on the ISA Server. The ISA Server decrypts the request. ISA Server sends an unsecured request to port 80 on the Web server. ISA Server receives the unsecured response from the Web server. ISA Server encrypts the response, and returns it to the client. This technique is useful in reverse-proxy situations, enabling ISA Server to require secure communications between itself and a client on the public network without the need to secure internal communication.

The common factor in these scenarios is that all client and server connections are made with ISA Server. Client and Web server never form connections between one another. Because SSL connections require greater setup effort than plain TCP connections, ISA Server is working a bit harder when it has to support secure connections under this scenario. With large numbers of connections, ISA Server may be working a lot harder.

The behavior of ISA Server with regard to SSL communication is configured with Web publishing rules for incoming requests and with routing rules for outgoing requests. Both Web publishing rules and routing rules can specify whether requests should be forwarded with SSL security.

**FIGURE 13**

Scenarios that employ SSL bridging.

To make SSL support even more versatile, Web publishing rules and routing rules can redirect requests intended for one server to another. Suppose, for example, the client sends the request to `http://www.pseudo-corp.com`, a Web server at another site in the user's organization. This server contains proprietary data, and all Internet communications with the server must use SSL. To redirect the request, a routing rule is defined that specifies requests to `www.pseudo-corp.com` are to be redirected to `wwws.pseudo-corp.com`. ISA Server transparently opens a secure connection with the target server, but communicates with the client using unsecured HTTP, which follows scenario 1 as discussed earlier in this section. Users don't need to remember or even be aware that the server is secure. ISA Server ensures that external communications with the secure server are always encrypted.

VPN Configuration Support

Some ISA Server functions are extensions of RRAS. One place where this is evident is ISA Server's support for Virtual Private Networks. VPN servers and clients can be configured within the ISA Server administration console. One of the interesting features is that the administrator who creates the VPN connection on one end can generate a file that provides input required to generate the connection at the other end. This greatly reduces the likelihood that the connection will be misconfigured. After the VPN connection is created in the ISA Server console, it is managed using the RRAS console.

Managing ISA Server

As you have probably realized already, ISA Server management is not a simple topic, so we have many things to discuss. After discussing installation, we'll move on to management issues.

Installing ISA Server

Before we can discuss ISA Software management procedures, we need to install it. A busy ISA Server has some fairly demanding hardware requirements, particularly in terms of memory, so we need to do some planning before we go too far. We'll start with server hardware requirements. Then we'll look at array planning. After that we can perform the actual installation.

Planning for Installation

Hardware requirements are the biggest planning issue we need to consider. If ISA Server is running with insufficient hardware resources, you won't like ISA Server one bit. If resources are adequate, ISA Server can actually speed up access to resources. Thanks to its caching capability, many client requests can be satisfied without the delay of communicating with a remote server. By reducing WAN traffic, ISA Server conserves WAN bandwidth, which should accelerate network-based processes that don't benefit from caching.

Planning Server Hardware

Microsoft's recommended minimum configuration for a computer running ISA Server is as follows:

- Pentium II or later processor, 300 MHz or better
- 256 MB RAM
- 20 MB available hard-disk storage in an NTFS partition (not required for firewall-only installations)

ISA Server is fairly memory intensive due to its need to manage large numbers of client-server connections and to buffer cache objects that are queued for disk storage. In testing, I've been able to get away with 128 MB of RAM, but that will only work in a very low-demand situation. Memory is probably the single most important hardware component where ISA Server performance is concerned. Invest in adequate memory before you pursue a processor upgrade. And, because hard disk space is cheap these days, be sure you have enough disk capacity. Insufficient disk capacity means that fewer objects can be cached, which will have an impact on performance.

The minimum configuration needs to be enhanced as the firewall component is required to handle higher traffic levels. Here are Microsoft's suggestions for processor enhancements required for various levels of traffic:

- A 300 MHz processor should support throughput up to 10 Mbps, as provided by ISDN, cable modem, DSL, T1, and other connections with limited bandwidth.
- A single 550 MHz processor should support throughput up to 50 Mbps, as provided by T3 service.
- An additional 550 MHz processor should be added for each additional 50 Mbps of traffic.

In all cases, when extra processors are required, they can be provided either by filling another processor slot on a multi-processor server or by adding servers to the ISA Server array. All things being equal, I prefer to expand the array. Adding more members to the array improves fault tolerance. A multi-processor server increases the risk associated with failure of a single server.

Two processors never perform twice as well as one. Some performance is always lost due to the requirement of coordinating the processors. Unfortunately, I don't have any test results that determine how ISA Server performance compares between a multi-server array and a single server with the same number of processors. Because the array expands all resources, however, I suspect an array will perform somewhat better for a given level of server hardware. Each server in the array adds an independent hard drive and network cards to the overall configuration. Some additional internal traffic will be generated by the protocol that coordinates cache operations among the servers in the array.

Unless hard drives and network adapters in a multi-processor server can keep up with the CPU, they will become a performance bottleneck that limits the usefulness of an additional processor. A busy multi-processor ISA Server should be equipped with fast hard drives, probably running on a hardware-based RAID array. This may be an instance when RAID level 0 (striping without parity) would be an appropriate choice. Without parity, nearly all of the available disk capacity is available. Parity slows down storage and provides little benefit for ISA Server since

the objects being cached are ephemeral. Because objects with dynamic content are not cached, nothing of great value can be lost if a disk in a RAID 0 array fails. Of course, RAID 0 arrays crash entirely if any disk in the array fails, whereas RAID 1 (mirroring) and RAID 5 (stripe sets with parity) have fail-safe mechanisms that keep the volume operating if a single disk drive fails. You may choose to sacrifice speed or capacity for the benefit of a fail-safe disk array.

Microsoft expresses requirements for forward caching in terms of the number of users that must be serviced:

- A single 300 MHz processor will support up to 250 users. 128 Mbps of memory and 2–4 GB of disk space should be added to the basic configuration for the first 250 users.
- A 550 MHz processor will support up to 2,000 users. Add 256 MB of RAM and 10 GB of disk storage to the basic configuration to support up to 2,000 users.
- For each additional 2,000 users add one 550 MHz processor, 256 MB of RAM, and 10 GB of disk space.

So one approach to scaling would be to add a 550 MHz server with 512 MB of RAM and 10 GB of disk space for each 2,000 users that must be supported.

Hardware recommendations for server publishing are expressed in terms of hits per second. Microsoft's recommendations are as follows:

- A single 300 MHz processor will support up to 500 hits/second. Add 128 MB of RAM to support requests at this level.
- A single 550 MHz processor will support up to 900 hits/second. Add 256 MB of RAM to support this number of requests.
- For each additional 800 hits/second, add one 550 MHz processor and 256 MB of RAM.

Planning the Local Address Table

When you install ISA Server in firewall or integrated modes, you will define a *Local Address Table*, referred to as a LAT. The LAT is a list of all the IP addresses that appear on the private network serviced by ISA Server. As far as firewall clients are concerned, if an IP address isn't in the LAT, it doesn't exist on your private network. Before installing ISA Server, you must determine all the IP address ranges on your private network so that the addresses can be included in the LAT.

When a firewall client needs to send a packet it determines whether the destination IP address is included in the LAT. If the IP address is in the LAT, the client sends the packet directly to the destination computer or through a route that is defined in the client's routing table. If the IP address isn't included in the LAT, the firewall client forwards the packet to its ISA Server.

NOTE

SecureNAT clients neither use nor have copies of the LAT. The ISA Server is configured as the client's default router, and any packets the client cannot send directly to the destination are sent to the ISA Server for routing.

Web Proxy clients receive a table of local addresses that is separately defined for the Web Proxy service.

Because ISA Server completely isolates your private network from direct communication with the Internet, you can use non-routable IP addresses, also known as private IP addresses, on the private network. When you install ISA Server, it is easy to include the following address ranges in the LAT:

- Class A: 10.0.0.0 through 10.255.255.255
- Class B: 172.16.0.0 through 172.31.255.255
- Class C: 192.168.0.0 through 192.168.255.255

The local address table is stored in a file named `msplat.txt`, which on the server is stored by default in the folder `C:\Program Files\Microsoft ISA Server\CLIENTS.Firewall and Web Proxy`. Firewall and Web Proxy clients periodically download the `msplat.txt` file.

NOTE

The `msplat.txt` file can be edited directly. I recommend you use WordPad rather than Notepad to edit the file. Notepad does not format the file for easy editing.

Entries in `msplat.txt` take the form

<first address of range> <TAB> <last address in range>

Here's an example:

```
172.16.0.0      172.31.255.255
```

The `msplat.txt` file contains entries that aren't displayed in the ISA Server console. Specifically, there is an entry for the range of addresses above the Class C range (224.0.0.0 through 255.255.255.254) and another range to cover loopback addresses (127.0.0.0 through 127.255.255.255). Don't remove or modify these entries.

Planning the Local Domain Table

Because Windows 2000 Active Directory names follow the DNS domain name format, a client cannot distinguish local domains from public domains by the name format alone. (The task is

easy with Windows NT because domain names don't include periods.) The Local Domain Table (LDT) lists domain names (domains and hosts) that the client is to regard as local.

Only firewall clients use the LDT, periodically downloading the LDT from ISA Server. When clients need to resolve a name, they consult the LDT. If the name does not appear in the LDT, the client directs the name query to ISA Server, which conducts the query on the client's behalf, returning the result. The client then consults the LAT to determine whether the IP address returned by ISA Server is local or remote.

The LDT is defined after installation is complete, but you should still plan ahead. The important thing about the LDT isn't really to identify local domains. It's to let the client know whether they can use a local DNS server to resolve names in the domain. If the DNS server is outside the firewall, the domain shouldn't appear in the LDT since that would prevent the client from directing queries to ISA Server.

If a domain is split with portions behind different ISA Server arrays, it should be regarded as local only if each network area has a local DNS server with a copy of the zone in which the domain is defined. Otherwise, clients must ask ISA Server to resolve the name using external DNS servers.

Planning the Network Architecture

Until now, we've only looked at very simple deployment plans for ISA Server, such as the one in Figure 14. Here, all computers are placed behind the same ISA Server array. Access rules permit internal clients to access outside servers. Publishing rules permit outside clients to access inside servers. Thus, the firewall permits two-way access.

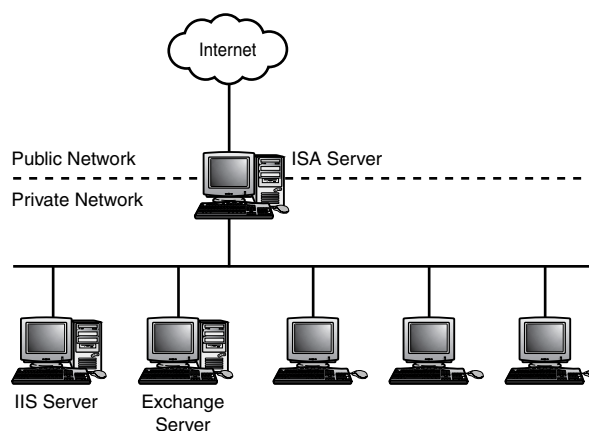


FIGURE 14

A single-level ISA Server architecture.

Some administrators would consider such a deployment undesirable because it grants external users access to the network where all computers are located. Although Web and server publishing rules should restrict those users' access to specifically delegated servers, there is always the chance that an outsider may find a way to bypass the restrictions and gain access to resources that shouldn't be shared publicly.

To enhance security, it is common practice to implement two private networks: one, called the *perimeter network* supports servers outsiders are permitted to access, and a second network supports private servers and clients and blocks all or most incoming service requests. Let's look at two methods of implementing those two networks. Microsoft calls these approaches the *back-to-back perimeter network configuration* and the *three-homed perimeter network configuration*.

The Back-to-Back Perimeter Network Configuration

The essential characteristic of this configuration is that there are two levels of firewall protection. Figure 15 shows the basic architecture. This configuration requires chaining. ISA Array 2 must be configured to forward outbound service requests to ISA Array 1.

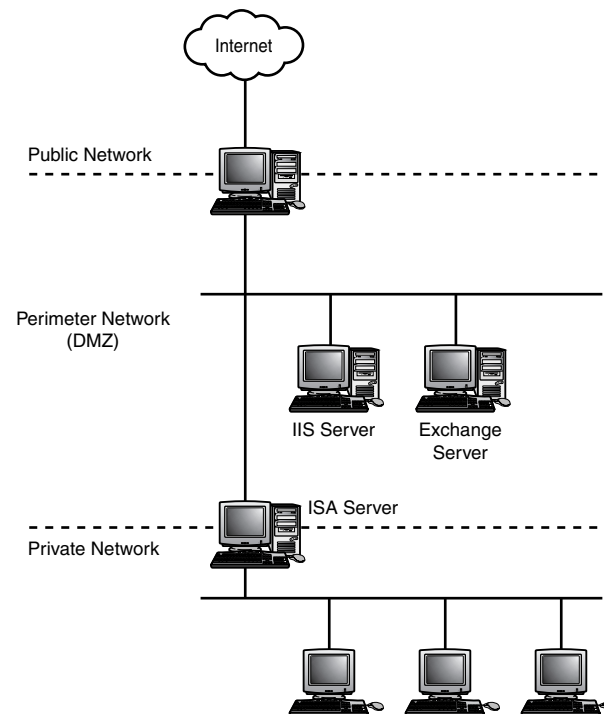


FIGURE 15

A back-to-back perimeter network ISA Server configuration.

Between the firewalls is a network that is sometimes referred to as the *perimeter network* but is more commonly called the *DMZ* (for de-militarized zone). All servers that are accessible by the public are connected to the DMZ.

The DMZ is locked up less tightly than the internal network. For example, the DMZ firewall might be configured with the following rules and filters:

- A protocol rule allows outgoing requests for HTTP, HTTPS, Gopher, FTP, and DNS. This rule enables internal clients to access outside Web servers and to query outside DNS servers.
- One or more site and content rules define the conditions for permitting user access to external Web servers.
- Packet filters allow outgoing DNS queries and ICMP echo requests, as well as select incoming ICMP replies. (These are the default ISA Server packet filters.)
- A Web publishing rule permits incoming access to IIS Server 1.
- A server publishing rule permits incoming access to Exchange Server 1 via SMTP and any required client protocols (for example, POP3, IMAP4, and/or MAPI). This rule enables external mail servers to send mail to Exchange Server 1. It also permits outside users to retrieve mail.
- A protocol rule allows outgoing SMTP traffic, enabling Exchange Server 1 to send mail to outside mail servers.
- A server publishing rule allows incoming DNS queries to the DNS server that is authoritative for pseudo-corp.com.

The ISA Server array that is protecting the private network would be configured along these lines:

- A protocol rule allows outgoing requests for HTTP, HTTPS, Gopher, FTP, and DNS.
- One or more site and content rules define the conditions for user access to external Web servers.
- A protocol rule allows inside users to send mail to and receive mail from Exchange Server 1.
- Packet filters allow outgoing DNS queries and ICMP echo requests, as well as select incoming ICMP replies. (The default filters.)
- Server publishing rules allow Windows 2000 Servers in the DMZ to request services from domain controllers on the internal network. Unless servers on the DMZ are stand-alone servers with their own user account databases, they must communicate with domain controllers on the internal network to authenticate users. The rule must allow incoming packets for Kerberos v5 (TCP and UDP ports 88 and 749) and RPC (TCP port 135).

- If any of the Windows 2000 Servers are domain controllers, the firewall must have a protocol rule that allows outgoing RPC and possibly Kerberos v5.
- If the DNS servers supporting Active Directory domains are all located on the private network, a server publishing rule must permit incoming DNS queries (UDP port 53) from servers on the DMZ. This protocol support is enabled in the Web access protocol rule that is configured when ISA Server is installed.

The only incoming service requests that are permitted enable Windows 2000 Servers to communicate between the DMZ and the private network. If ISA Server is functioning in an array configuration, its computer must be a domain member. It must therefore be able to communicate with DCs in the domain. The protocols used, particularly RPC, can be blocked by packet filters in ISA Array 1.

Routing must be configured carefully on this sort of network. Clients on the private network communicate in only two ways: through local routing or through ISA Array 2. Consequently, firewall clients can be configured through the LAT and LDT.

SecureNAT clients will use ISA Array 2 as a default router. ISA Array 2 can route packets back to the private network if there are multiple subnets. Alternatively, a separate router can be configured on the private network and used by local computers as their default router. The router should have dynamic or static routes to local subnets and should use ISA Server 2 as its default router.

Web Proxy clients will use local address and domain tables that are copied from the Web Proxy service on ISA Server 2. The Web Proxy client configuration causes the client to direct requests for outside services to Web Proxy 2.

Clients on the perimeter network use configuration information that is supplied by ISA Array 1. One approach to routing is to include addresses on the perimeter and private networks in the perimeter network LAT. If a server on the perimeter network is configured as a router, Firewall and Web Proxy clients can use the router as their default router. This router could direct packets to the perimeter network and to the private network via ISA Array 2.

SecureNAT clients rely entirely on routing to direct packets to destinations. Clients on the perimeter network could be configured with static routes to networks on the private network, or all routing could be performed by using ISA Server 1 as a default router. ISA Server 1's routes to the private network can be configured using static routes or by a dynamic routing protocol.

If dynamic routing protocols are used in any of these situations, routers on the private and perimeter networks must be able to communicate. ISA Router 2 must be configured with packet filters that allow routing protocol packets to be forwarded.

Should servers on the DMZ network be configured as Windows 2000 Server domain controllers? Doing so would enable the servers to maintain a local copy of the domain database, which in turn enables them to authenticate clients without the need to consult another server. However, it also places services on the DMZ that we ordinarily want to keep private. Of course, the DMZ firewall shouldn't publish Kerberos v5 or RPC services to the Internet, and that alone makes the Windows 2000 Servers less vulnerable. Depending on the security level that is desired, however, it may be preferable to keep all domain controllers on the private network.

Still, if you want to be especially careful with any domain controller in the DMZ, there are some things that you can do to tighten its security:

- Define input and output filters that block Kerberos v5 and RPC packets that are not addressed from the private network (actually, the ISA Server that protects the private network).
- Disable NetBIOS if it is not needed. Otherwise install packet filters that block NetBIOS packets on the ISA Server interfaces that connect to the Internet.
- Run a port scan on the server and block any ports not required for the intended server functionality.

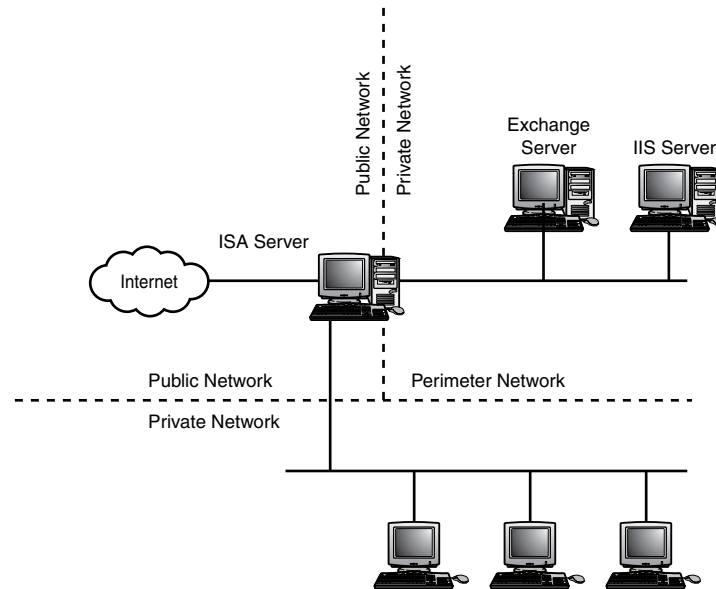
If the expense can be justified, back-to-back is probably the preferred configuration. It can be easier to maintain than the three-homed configuration, which can't use publishing rules. And it provides a clear security hierarchy. The private network can't be less secure than the peripheral network, so securing the private network is simply a matter of blocking packets on the peripheral network so they can't enter the private network.

The Three-Homed Perimeter Network Configuration

The three-homed configuration can be accomplished with a single ISA Server or array. Figure 16 illustrates the setup for a single ISA Server. If the ISA Server is a member of an array, each computer in the array must have three interfaces and be connected in the same manner.

The ISA Server is equipped with three network interface adapters, with one interface connected to the public, perimeter, and private networks. If an ISA Server array is deployed, each server in the array must be equipped with three interfaces that connect to each of the three networks.

Access rules determine the types of service requests ISA Server allows to reach the Internet. That much of the configuration follows procedures we've already examined.

**FIGURE 16**

A three-homed perimeter network ISA Server configuration.

However, in a three-homed configuration, access to servers on the peripheral network cannot be controlled using publishing rules. Instead, packet filters must be defined for each service to be published. The packet filters have these properties:

- The filter type is Custom.
- The IP protocol type is either TCP or UDP as required for the service.
- The local port number is the port required for the service.
- The remote port is All Ports.
- The direction is either Both (for TCP) or Receive send for UDP.

We'll look at these packet filter properties in detail later in the section "Defining IP Packet Filters."

NOTE

If you look up this topic in ISA Server Help (at least with Release Candidate 1) the filter type required for this network configuration is specified as "Open." There is no such filter type. Create a custom filter instead.

Installation

After you have done the planning suggested in the foregoing sections, installing ISA Server is not at all difficult. Start by executing ISAAutorun.exe from the root of the CD-ROM. If autorun is enabled for the drive, ISAAutorun.exe is executed when you insert the CD-ROM. The **Microsoft ISA Server Setup** dialog box will be displayed, as shown in Figure 17.



FIGURE 17

The ISA Server Setup dialog box.

ISA Server installation requires two procedures:

1. The Active Directory must be extended with objects required by ISA Server.
2. After configuring Active Directory, ISA Server can be installed.

Extending the Active Directory Schema

To install the Active Directory schema extensions, you must be a member of the Enterprise Admins and the Schema Admins groups. The built-in Administrator user account is initially configured as a member of both groups. Because Active Directory is modified, the procedure must be executed on a Domain Controller, even though ISA Server will not be installed on that computer. The modifications to the AD schema cannot be reversed, but that's not a problem. The schema extensions are simply ignored if all ISA Servers are decommissioned.

The procedure is simple:

1. Click **Run ISA Server Enterprise Initialization** in the **ISA Server Setup** dialog box.
2. Click **Yes** in the **ISA Server Enterprise Initialization** dialog box.
3. Configure settings in the **ISA Enterprise Initialization** dialog box (Figure 18) as follows:
 - **Use array policy only.** Select this option if you want to configure each ISA Server array independently and without an enterprise policy.
 - **Use this enterprise policy.** Select this option if you want to configure one or more Enterprise policies. When this option is enabled and **Also allow array policy** is not selected, only the enterprise policy is used to configure protocol rules and site and content rules. If you select this option you must specify a name for the first enterprise policy, which must be configured after ISA Server is installed. During ISA Server installation, an initial enterprise policy is created with the default name **Enterprise Policy 1**.
 - **Also allow array policy.** If enterprise policies are permitted, check this box to permit array policies to modify rules defined by the enterprise policy. Allowing array policies enables management to be distributed so that different managers can adjust the settings in their ISA Server arrays to match local or departmental requirements. If array policies are not permitted, the enterprise policy determines all protocol rules and site and content rules for the array. Array publishing rules and packet filtering must be explicitly allowed as described in the next two bullets.
 - **Allow publishing rules to be created on the array.** This option must be selected to allow publishing rules to be defined for arrays. If the option is not enabled, the array is restricted to processing outgoing requests only. Publishing rules enable outside clients to communicate with servers inside the ISA Server array.
 - **Use packet filtering on the array.** This option must be selected to allow packet filters to take effect. Packet filters are created only at the array level. Deselect this option if the array is to be configured only by access rules and by publishing rules if publishing rules are enabled for the array.
4. The setup program now adds ISA Server schema extensions to Active Directory. The procedure requires several minutes. You are notified when the updates are complete.

All of the properties in the **ISA Enterprise Initialization** dialog box can be modified after installation. This procedure defines default properties that function at the enterprise level. These default properties can be overridden at the enterprise.

**FIGURE 18**

Configuring Enterprise Properties when initializing the enterprise.

Installing the ISA Server Software

After installing the AD schema extensions, ISA Server can be installed on any server in the forest because all trees in the forest must have the same schema. Perform this procedure on the computer that will be used as the ISA Server:

1. Install Windows 2000 Service Pack 1 on the target computer.
2. Open the **ISA Server Setup** dialog box as described in the previous section and click **Install ISA Server**. Alternatively, run `\isa\setup.exe` on the CD-ROM.
3. Skip the introductory dialog box, enter your CD key, agree to the license, and bow in the general direction of Redmond, Washington to reach the first useful dialog box, shown in Figure 19. The ISA Server product consists of three components:
 - **ISA Services** are the ISA Server software.
 - **Administrative tools** consist of the MMC snap-ins and other modules that are used to manage ISA Server. The administrative tools can be installed on the ISA Server itself or on another computer that will manage the server remotely.
 - **Add-in services** include the H.323 Gateway Service, a protocol that is used by NetMeeting. Also included is the Message Screener component that performs content filtering on incoming SMTP traffic. Neither of these components is considered in this chapter.
4. Select an installation option:
 - **Typical Installation** installs ISA Services and Administrative tools.
 - **Full Installation** installs all components. (Duh!)
 - **Custom Installation** enables you to select the options to be installed. For example, use this option to add the H.323 Gateway Service to an existing configuration. Or use it to configure a computer with only the ISA Server administrative tools.

**FIGURE 19**

The ISA Server Setup dialog box.

By default, ISA Server is installed in the folder `C:\Program Files\ISA Server`. Click **Change Folder** to select another destination.

5. If no arrays are currently defined and you are installing ISA Server on a computer that is a Windows 2000 domain member, the next dialog box is **New Array**. Specify a name for the array. Calling on my special creative flair I named mine Array 1.
6. The next dialog box offers three choices that specify the installation mode:
 - **Firewall mode.** Select this option to install only the SecureNAT and enterprise firewall components.
 - **Cache mode.** Select this option to install only the cache and Web Proxy components. Cache mode requires at least 20MB of available space on an NTFS storage volume.
 - **Integrated mode.** Select this to install SecureNAT, enterprise firewall, cache, and Web Proxy components. Integrated mode requires at least 20MB of available space on an NTFS storage volume.
7. The next dialog box, shown in Figure 20, is used to configure the cache. These settings can be modified after installation. The best performance can be had by placing the cache on a RAID storage array or by distributing cache space across multiple separate SCSI or Fibre Channel disks. Little or no benefit is gained by distributing the cache across multiple EIDE disks.

Cache can be allocated on any NTFS volume as follows:

 - a. Select a volume in the **Drive** list. Below the **Drive** list, the characteristics of the volume will be listed, informing you of the space that is available on the drive.
 - b. Edit the **Cache size (MB)** field to specify the size of the cache on this volume.
 - c. Click **Set**.

Repeat this procedure for other volumes that will contain ISA Server cache. The total size of the cache is displayed after the **Total cache size (MB)** label. Cache performance is improved by distributing it across multiple SCSI hard drives.

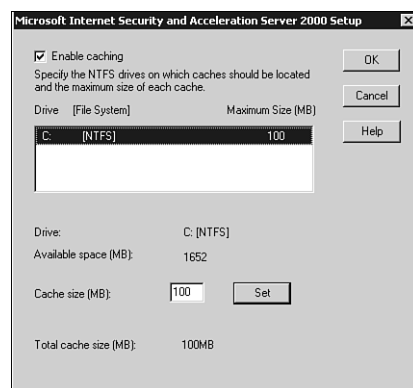


FIGURE 20

The initial cache configuration is specified during installation.

8. The next step is to configure the LAT with the dialog box shown in Figure 21. You can define addresses manually or select them from a table that includes the private IP address ranges as well as IP addresses derived from the computer's routing table.

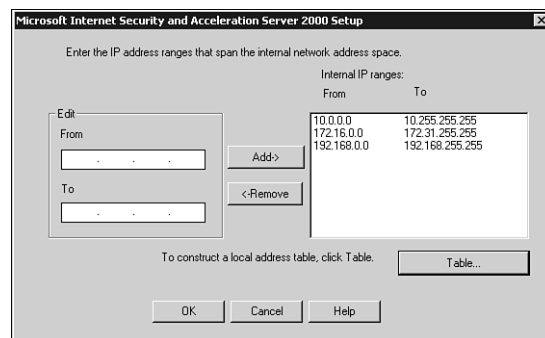


FIGURE 21

This dialog box defines initial entries in the LAT.

To define an address range manually:

- a. Enter the first address in the range in the **From** field.
- b. Enter the last address in the range in the **To** field.
- c. Click **Add** to copy the address range into the **Internal IP ranges** list.

To copy private IP address ranges and ranges based on the computer's routing table, do this:

- a. Click **Table** to open the **Local Address Table** dialog box shown in Figure 22. (Initially the dialog box contains no data. In the figure, I have added the private address ranges to the LAT.)
- b. Check the top check box to include the private address ranges 10.0.0.0/24, 172.16.0.0/12, and 192.168.0.0/16.
- c. Check the second check box to include address ranges derived from the computer's routing table. Then check any or all of the address ranges in the following list.
- d. Click **OK** to add the selected ranges to the LAT.

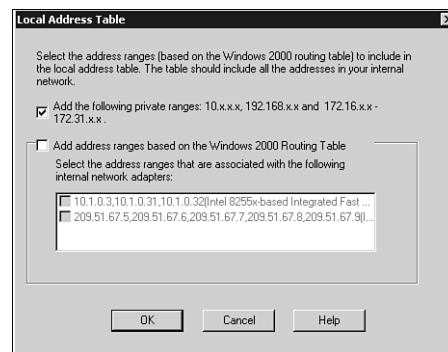
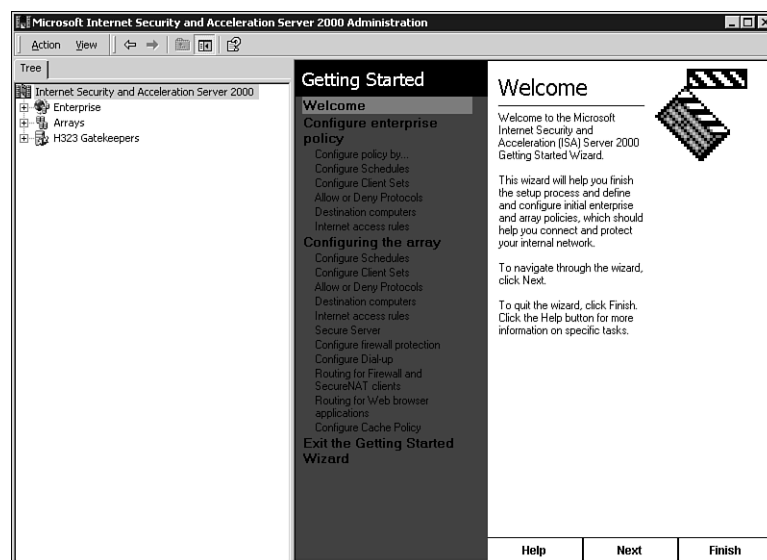


FIGURE 22

Selecting LAT entries from private IP address ranges and address ranges derived from routing tables.

9. Installation begins when you leave the LAT definition dialog box. In the process of installation, IIS will be stopped and restarted if it is installed on the computer.
10. The next dialog box gives you the option of starting the **ISA Administrator Getting Started Wizard** (see Figure 23).

This wizard isn't like other Windows wizards you've encountered. It doesn't march you through a series of steps that, when completed, practically guarantee that you will have a properly configured component. Frankly, ISA Server is too complex to configure in such a linear manner. All the wizard does is provide a list of tasks that you can select to reveal options for that task. However, the order in which the tasks are listed is not necessarily the order in which you will want to perform them, and some tasks you need to perform aren't accessible through the wizard. Since everything done in the wizard can be accomplished using the ISA Server console, and some vital functions can't be started from the wizard, I won't follow the wizard at all. Simply click **Finish** at the lower-right corner of the wizard and proceed to the next section.

**FIGURE 23**

The ISA Administrator Getting Started Wizard is one option for configuring ISA Server.

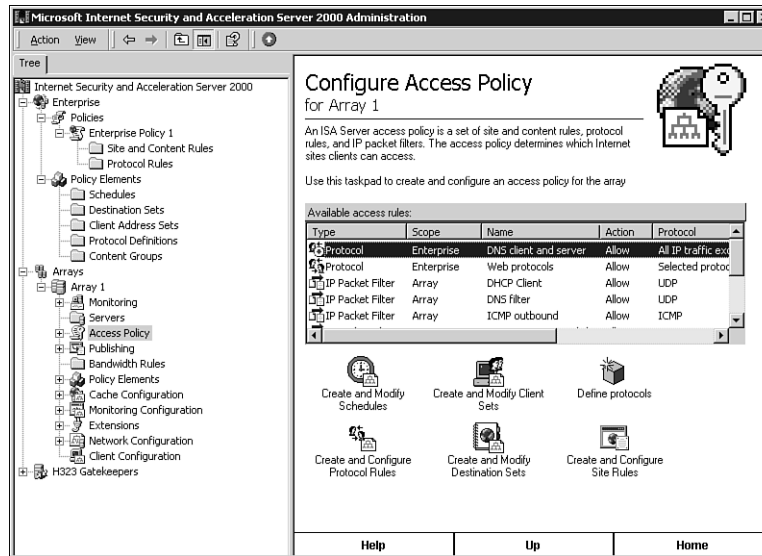
You can start the wizard from the console by selecting **Internet Security and Administration Server 2000** in the console object tree and then clicking **Getting Started Wizard** in the right-hand pane. The console must be in Taskpad mode, which is explained in the next section.

The ISA Server Administration Console

The ISA Server console offers two ways of working with leaf objects:

- A conventional Details view that is familiar from most MMC snap-ins, where objects are listed in the right pane and most actions are performed by right-clicking an object and choosing options from a context menu. To select Details view, open the **View** menu and click **Taskpad** to remove the check mark.
- A new Taskpad view displays objects in a more graphically oriented manner, substituting icons for many context menu commands. To select Taskpad view, open the **View** menu and click **Taskpad** to check it.

In Figure 24, I've configured the ISA Server console in Taskpad view and selected the object **Arrays**→**Array 1**→**Access Policy**→**Access Policy** in the object hierarchy. The Taskpad for this object has features that are found in most Taskpads. A scrolling table lists all of the objects that have been defined for the selected container object. Most tasks are initiated by clicking one of the icons in the Taskpad.

**FIGURE 24**

The ISA Server console Taskpad view uses icons to replace many context menu functions.

The **Up** button at the bottom of the Taskpad moves the selection up in the object tree to the object from which the current object was accessed. For example, the Taskpad for **Enterprise** contains a button named **Create and Configure Enterprise Policies**, which branches to a Taskpad by the same name. If you click **Up** in that Taskpad, your viewpoint reverts to the **Enterprise** object. Sometimes the results of clicking **Up** are unpredictable, and unless you know which Taskpad icon links to another Taskpad, the results you get by clicking **Up** may be difficult to anticipate.

The **Home** button moves the selection to the first-level object that contains the current object, such as **Enterprise** or **Arrays**. I'm not sure what value this function adds to the console, but it's there.

I'm not a big fan of Taskpad view. It seems to enforce the suspicion that Microsoft is never happy with a clean, substantially text-based interface such as the Details pane when it can have big splashy icons instead. My biggest complaint with Taskpads has to do with the object lists that appear in the Taskpad. While you can adjust the columns in the list, they revert to their original settings whenever the list is refreshed, which happens often. You will almost always need to expand the first column to see the object's entire name, but it never stays expanded very long. Also, unless they are fixed before ISA Server ships, the Taskpad conventions are a tad inconsistent. Finally, there are a few functions that can only be performed in Details view, requiring you to switch back and forth. Give me the Details view any day.

Most of my remaining figures will use Details view. When the two views require distinct methods to start a task, I will describe both methods.

In Figure 24, I've expanded the tree in select places. Specifically, I've expanded the **Policies** and **Policy Elements** objects under **Enterprise**. Notice that the open objects under **Array 1** contain elements that don't appear under **Enterprise**. Many types of objects can be defined only at the array level.

Let's start looking at details, working from general to specific. Our first stop is the **Enterprise** subtree.

Defining the Enterprise

All ISA Servers in a given Active Directory tree belong to an Enterprise. Within the **Enterprise** container administrators can define **Enterprise Policies**.

Figure 25 shows the general organizations of Enterprise- and Array-related objects with the **Enterprise** object selected. At least one **Enterprise Policy** object will be defined, but administrators can create as many **Enterprise Policy** objects as are necessary. Each **Enterprise Policy** object can be used to configure one or more ISA Server arrays, although it is possible to configure things so that they will not be affected by an enterprise policy.

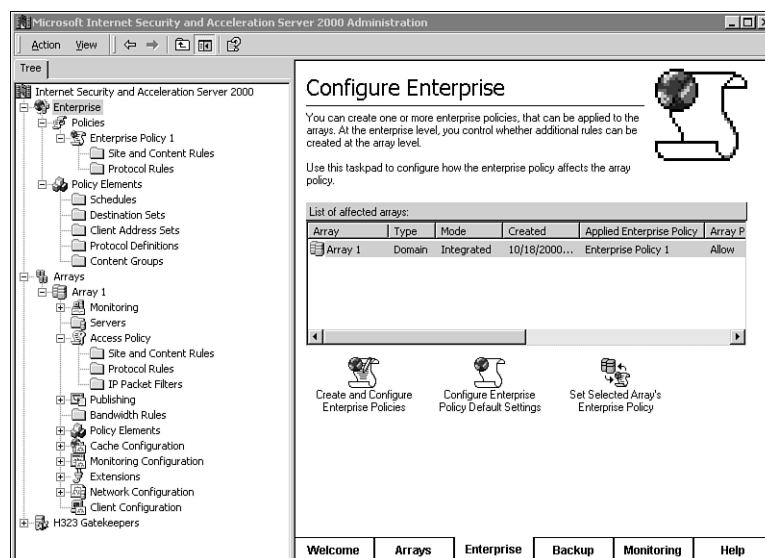


FIGURE 25

The ISA Server console, displaying the Taskpad for the Enterprise object.

Use and relationships of the **Enterprise Policy** and the array **Access Policy** can be configured in three ways:

- All access policy and policy element properties can be defined by the enterprise policy associated with the array. This approach centralizes and tightly controls array management. The enterprise administrator can determine whether arrays in the enterprise can publish services and whether or not to force packet filtering on the array.
- The enterprise policy can be ignored so that all access policy and policy element properties are defined at the array level. This approach distributes array management and eliminates most central administration.
- The enterprise policy can provide initial rule definitions for the array. The array inherits access policy rules and policy elements from the enterprise. Inherited rules and policy elements can be further restricted at the array level but cannot be less restrictive than limits established in the enterprise policy. This approach balances central authority with local autonomy and is probably the most commonly used approach.

Default enterprise policy settings are defined under the **Enterprise** object. Let's look at specific **Enterprise** object management procedures.

Configuring Enterprise Policy Default Settings

To define the enterprise policy default settings, do one of the following:

- In Details view open the **Set Defaults** dialog box by right-clicking **Enterprise** in the object tree and choosing **Set Defaults** in the context menu.
- In Taskpad view, click the button **Configure Enterprise Policy** to open the **Default Settings** dialog box.

We saw an identical dialog box when installing ISA Server. See the discussion relating to Figure 17 for information about these settings.

NOTE

Site and content rules, protocol rules, and five types of policy elements can be defined at both the enterprise and array levels. The procedures for creating and managing these objects is the same at either level, so I won't repeat this information for every object type.

Just remember that the enterprise policy and the array access policy determine which policies are active and how they relate. If the Enterprise Policy disables array policies, you can still define site and content rules, protocols and policy elements in the containers below the **Array** object. However, these properties will have no effect on ISA Server behavior. You can't figure out the active configuration for ISA Server without taking the Enterprise Policy into account.

The default Enterprise policy settings are determined by a particular Enterprise Policy. To see how to modify the defaults and to configure additional enterprise policies, we need to examine the creation and configuration procedures for enterprise policies.

Creating and Configuring Enterprise Policies

There will always be one **Enterprise Policy** object, which by default is named **Enterprise Policy 1**. To modify or create enterprise policies, click the **Create and Configure Enterprise Policies** in the **Configure Enterprise** Taskpad.

The **Create and Configure Enterprise Policies** Taskpad is shown in Figure 26. This Taskpad can also be accessed by selecting the **Enterprise**→**Policies** object.

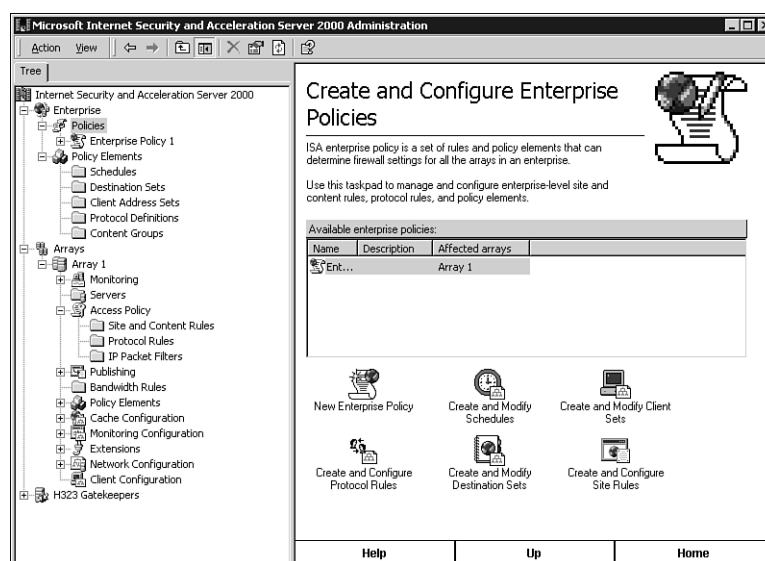


FIGURE 26

The Create and Configure Enterprise Policies Taskpad.

Several icons in this Taskpad access Taskpads used to create other types of objects. We won't examine the links for all Taskpad icons, but it is useful to examine the ones appearing in the **Enterprise Policies** Taskpad:

- **Create and Modify Schedules.** This button opens the Taskpad used to create Schedules policy elements.
- **Create and Modify Client Sets.** This button opens the Taskpad used to create Client Address policy elements.

- **Create and Modify Destination Sets.** This button opens the Taskpad used to create destination set policy elements.
- **Create and Configure Protocol Rules.** This button opens the Taskpad used to create protocol rules.
- **Create and Configure Site Rules.** This button opens the Taskpad used to create site and content rules.

The only procedures we will examine in detail are those of creating and modifying enterprise policies.

Creating New Enterprise Policies

You may need more than one enterprise policy to provide base configuration properties for arrays at different locations or having different functionality. New enterprise policies are created as follows:

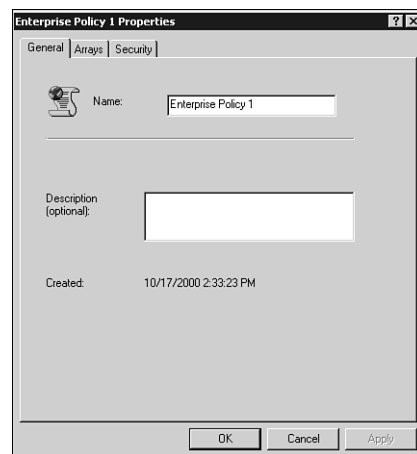
1. To start the **New Enterprise Policy Wizard**:
 - In Details view, right-click **Enterprise**→**Policies** and select **New**→**Policy** in the context menu.
 - In Taskpad view, select **Enterprise**→**Policies** and click the **New Enterprise Policy** icon.
2. In the first dialog box, enter a name for the policy.
3. The next dialog box offers two options:
 - **Create a new policy.** Select this option to create a policy from scratch. That's the course we will take through the wizard.
 - **Copy this policy.** Select this option to use an existing policy as a starting point. You must specify an existing enterprise policy, which can be selected from a drop-down list.
4. Finish the wizard. (Don't you wish everything with Windows 2000 was that simple?)

The new enterprise policy object will be added to the **Policies** container and will appear in the list in the Details pane or the Taskpad.

Configuring Enterprise Policies

New enterprise policies always require configuration as do many policies that are created as copies of existing policies. We'll turn our attention to that procedure next.

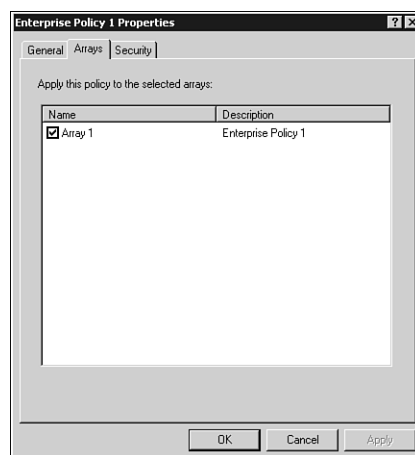
To configure an enterprise policy, double-click its entry in the **Enterprise Policy** Taskpad or Details panel. The **Enterprise Policy Properties** dialog box, shown in Figure 27, has three tabs.

**FIGURE 27**

Enterprise Policy Properties: The General tab.

The **General** tab includes descriptive information only. Edit the **Name** and **Description** fields as required.

The **Arrays** tab appears in Figure 28. All configured arrays will be listed. Check an array to associate it with this enterprise policy.

**FIGURE 28**

Enterprise Policy Properties: The Arrays tab.

The **Security** tab, shown in Figure 29, is used to determine the functions that users and groups can perform on the enterprise policy. This and similar dialog boxes for other objects can be used to stratify management authority for enterprise policies and access policies. You could deny a user the right to edit the enterprise policy in several ways:

- By removing the user from the Domain Admins and Enterprise Admins groups
- By adding the user's User object to the list and checking **Deny** for all permissions except possibly **Read**
- By adding the user to a group and denying all permissions for the group except possibly **Read**

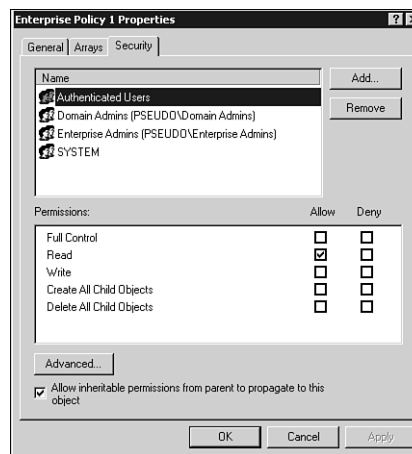


FIGURE 29

Enterprise Policy Properties: The Security tab.

Deny permissions take precedence over allow permissions, so a permission allowed in Domain Admins or Enterprise Admins is overridden by a deny permission assigned to the user's user account or to any group of which the user account is a member.

As you can see from the property pages, the enterprise policy object itself is very simple and has only two functions. It identifies the arrays that are configured by the enterprise policy, and it is a container for the access rules that define the capabilities and limitations of users accessing outside network services through ISA Servers in the array.

We'll return to enterprise policy rules in the sections "Defining Protocol Rules" and "Defining Site and Content Rules."

Modifying and Deleting Enterprise Policies

There are no pretty icons for modifying and deleting an enterprise policy. You have to start things using ugly old menus. To modify an enterprise policy, open the **Enterprise Policy Properties** dialog box using one of these procedures:

- In the object tree, right-click **Enterprise**→**Policies**→*enterprise policy name* and choose **Properties** in the context menu.
- Select **Enterprise**→**Policies**→*enterprise policy name* and double-click the enterprise policy in the Taskbar or Details panes.

To delete an enterprise policy use one of these procedures:

- In the object tree, right-click **Enterprise**→**Policies**→*enterprise policy name* and choose **Delete** in the context menu.
- Select **Enterprise**→**Policies**→*enterprise policy name*, select the enterprise policy in the Taskbar or Details panes, and press Delete.

Managing ISA Server Arrays

An ISA Server array consists of one or more ISA Servers that share a common configuration. Some properties can be defined for individual servers, but the servers are for the most part identical. If caching is enabled on the servers in an array, the servers communicate to establish a shared virtual cache that can be consulted by any of the servers.

The servers in an array are grouped under an array name. Firewall clients can be configured to access ISA Servers in the array as a group using an array DNS name. A service request that is directed from a firewall client to the array can be processed by any of the array members.

Upon first installation, ISA Server is configured with one array with the default name **ISA Array 1**. Subsequent ISA Servers can be added to an existing array or to a new array.

Array Functions

Arrays define far more ISA Server and client functionality than do enterprises. Besides protocol rules and site and content rules, arrays configure the following ISA Server capabilities:

- Dial-up entries and bandwidth priorities policy elements. See “Defining Policy Elements.”
- Publishing rules. See “Defining Web Publishing Rules” and “Defining Server Publishing Rules.”
- Web Proxy routing rules. See “Defining Web Publishing Rules.”
- Packet filters. See “Defining IP Packet Filters.”

- Cache configuration. See “Managing the Cache.”
- Bandwidth rules. See “Bandwidth Priorities.”
- Configuring firewall clients. See “Configuring ISA Server Clients.”

We’ll look at all those capabilities in the remaining sections. Before leaving this topic, however, we need to look at the properties that configure the array itself.

Array Properties

To view the properties pages for an array, right-click **Arrays**→*array name* and choose **Properties** from the context menu. The **General** tab for an array is shown in Figure 30.

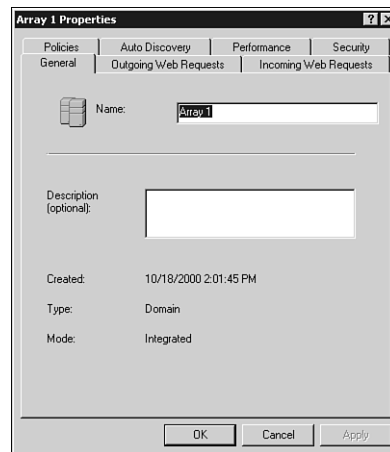


FIGURE 30

Array Properties: The General tab.

Array Properties: The General Tab

The **General** tab has only two configurable fields: **Name** and **Description**, neither of which requires explanation.

The **Type** will be either **Domain**, indicating that the server is a member of a domain-based array, or **Standalone**, indicating that the server is not a member of an array.

The **Mode** will be **Firewall**, **Cache**, or **Integrated**.

Array Properties: The Outgoing Web Requests Tab

The **Outgoing Web Requests** tab is shown in Figure 31. Properties on this tab determine how the ISA Server will respond to outgoing Web requests. The process consists of configuring *listeners* that receive outgoing Web requests. ISA Server can be configured to use a single

listener for all internal IP addresses, or to use multiple listeners that are selected for different internal IP addresses.

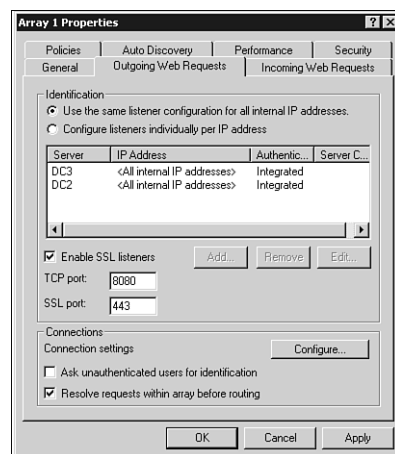


FIGURE 31

Array Properties: The Outgoing Web Requests tab.

To specify the listener configuration, select one of the following options:

- **Use the same listener configuration for all internal IP addresses.** With this option, a single listener is defined in the list box for each server in the array. This listener applies to all internal IP addresses on the server. This is the default setting.
- **Configure listeners individually per IP address.** With this option, a separate listener must be defined for each IP address that is assigned to the ISA Server's internal interface.

See the section “Configuring Web Listeners” for more information about listeners.

The following fields complete the **Outgoing Web Requests** tab:

- **Enable SSL listeners.** Check this box if the ISA Server is to listen to incoming HTTPS requests.
- **TCP port.** Specify the port the Web Proxy service will monitor for outgoing Web requests. The most commonly used port for Web Proxy clients is 8080.
- **SSL port.** Specify the port the Web Proxy service will monitor for outgoing secure HTTP requests. Commonly used ports for ISA Server clients are 443 and 8443. Many Web clients can only use port 443.

- **Connection settings.** Click **Configure** to open the **Connection Settings** dialog box. Complete it as follows:
 - **Unlimited.** Select this radio button if there is to be no limit to the number of outgoing Web connections the ISA Server or array will support.
 - **Maximum per server.** Select this radio button to limit the number of connections that each server will support at one time. Specify the number of connections in the field below this radio button.
 - **Connection timeout.** This property determines the time in seconds that a connection can be inactive before it is disconnected.
- **Ask unauthenticated users for identification.** This option requires all users to be authenticated using one of the methods that is enabled for the listener that accepts the client connection.
- **Resolve requests within array before routing.** Check this option to enable array members to pool their caches. If the option is not enabled, each server in the array will consult only its individual cache. The option is enabled by default.

Array Properties: The Incoming Web Requests Tab

Settings on this tab are identical to those on the **Outgoing Requests Tab**, although the defaults are different. Table 2 summarizes the default settings for outgoing and incoming Web requests.

TABLE 2 Default Settings For Outgoing and Incoming Web Requests

<i>Property</i>	<i>Outgoing Web Requests</i>	<i>Incoming Web Requests</i>
Use the same listener configuration for all internal IP addresses	Yes	No
Configure listeners individually per IP address	No	Yes
Enable SSL listeners	No	Yes
TCP port	8080	80
SSL port	Not defined	443
Connection settings	Unlimited	Unlimited
Ask unauthenticated users for identification	No	No
Resolve requests within array before routing	Yes	No

The differences noted in the table reflect the differing natures of outgoing and incoming requests. When multiple internal Web servers are published, outside users need to be able to address service requests to each. The most common technique is to multi-home the external interface of the ISA Server and define routing in the Web publishing tool that maps the external IP address to the internal IP address.

Configuring Web Listeners

As discussed, listeners listen for and process both outgoing and incoming Web requests.

If **Use the same listener configuration for all internal IP addresses** is selected, a single listener definition appears in the list for each server in the array. Select the listener and click **Edit** to open the **Add/Edit Listeners** dialog box where the listener can be modified.

If **Configure listeners individually per IP address** is selected, the list is initially empty, requiring you to create the required listener definitions. Use the **Add**, **Remove**, and **Edit** buttons to create, delete, and modify listener definitions.

Figure 32 shows the **Add/Edit Listeners** dialog box. The fields in the dialog box are

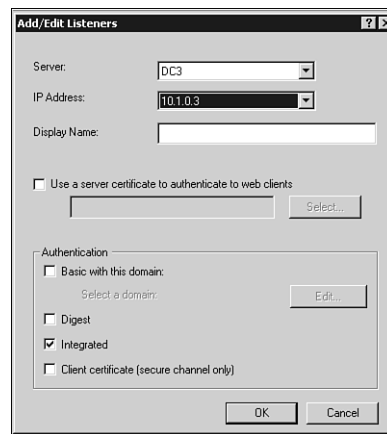
- **Server.** In the pull-down box, select the server for which the listener is defined. If **Use the same listener for all internal IP addresses** is selected, this field is pre-filled with the name of the server for which the listener is defined and cannot be modified.
- **IP Address.** In the pull-down box, select the IP address for which the listener is defined. If **Use the same listener for all internal IP addresses** is selected, this field is pre-filled with <All internal IP addresses> and cannot be modified.

NOTE

The **Server** and **IP Address** fields cannot be modified after the listener is created.

- **Display Name.** Enter the name that will identify the listener in the **Display Name** column of the list on the **Outgoing Web Requests** tab.
- **Use a server certificate to authenticate to web clients.** To authenticate the server with a certificate, check this box and click **Select** to choose a certificate from those that are installed on the computer.
- **Authentication: Basic with this domain.** Select this option to support unencrypted text-based authentication. Then click **Edit** and enter the name of the domain in which users are authenticated.
- **Authentication: Digest.** Select this option to support digest-based authentication.

- **Authentication: Integrated.** Select this option to support Window-integrated authentication.
- **Authentication: Client certificate (secure channel only).** Select this option to authenticate the client by a client certificate. This option applies only to secure communication via HTTPS.

**FIGURE 32**

This dialog box is used to create and modify Web listeners.

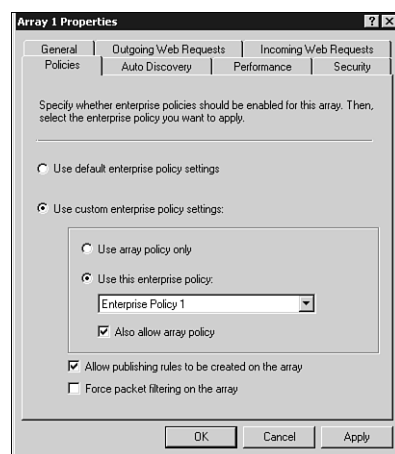
Array Properties: The Policies Tab

The **Policies** tab, shown in Figure 33, specifies the relationship of the array with enterprise policies. Settings in this dialog box are effective only if enterprise defaults (see “Configuring Enterprise Policy Default Settings”) permit the array to be configured separately from an enterprise policy.

Configure array policies as follows:

- **Use default enterprise policy settings.** Default enterprise policy settings are those defined in the enterprise policy that is selected in the **Enterprise Defaults** dialog box.
- **Use custom enterprise policy settings.** Select this option to define the array’s use of enterprise policies on a custom basis. When this option is selected these three fields can be modified:
 - **Use array policy only.** Select this option if array policy properties are independent of any enterprise policy. If this option is selected, arrays are unaffected by an enterprise policy and other options in the dialog box are deactivated.

- **Use this enterprise policy.** If an enterprise array will configure the array, check this option and select the name of the enterprise policy in the pull-down list. If **Use this enterprise policy** is selected and **Also allow array policy** is not selected, array policies cannot be configured. This option must be selected to activate the remaining options in the dialog box.
- **Also allow array policy.** This option is active only if **Use this enterprise policy** is selected. Check this option to permit the array policy to modify settings established by the enterprise policy. The array policy cannot be less restrictive than the enterprise policy.
- **Allow publishing rules to be created on the array.** If this option is not selected, the array will support protocol rules and Web access rules, but will not support Web or server publishing rules.
- **Force packet filtering on the array.** If this option is selected, the array must support packet filters.

**FIGURE 33**

Array Properties: The Policies tab.

Array Properties: The Auto Discovery Tab

Web Proxy and firewall clients must be configured with the address of the ISA Servers they are to use. Client configuration can be established statically, but it is usually preferable to enable clients to automatically discover ISA Server. Web Proxy clients discover Web Proxy servers using the Web Proxy Auto Discovery Protocol (WPAD). Firewall clients discover ISA

firewall servers via a similar WinSock-based protocol identified as WSPAD. ISA Server automatic discovery requires some configuration at the client, which will be discussed at the end of the chapter in the section “Configuring Web Proxy Clients.”

Automatic discovery is a great solution to the problems posed by mobile clients, which must be configured differently depending on the subnet they connect to and whether they are connecting from outside the firewall or inside the firewall. It is also a great way to roll out changes in the ISA Server architecture without the necessity of reconfiguring clients individually.

Client auto configuration must be enabled in the array properties on the **Auto Discovery** tab. This tab has only two properties:

- **Publish automatic discovery information.** Check this option to enable auto discovery.
- **Use this port for automatic discovery requests.** This field is active if automatic discovery is enabled, and specifies the internal port on which the Web Proxy service will listen for automatic discovery requests from clients. The default port is 80. As is mentioned frequently in this chapter, the Web Proxy service typically accepts requests from Web Proxy clients on port 8080. When they are initializing, however, clients using automatic discovery have yet to learn the port used to communicate with the Web Proxy service, so clients typically use the default HTTP port 80.

This procedure enables automatic discovery publishing on the ISA Server array, but additional configuration steps are required for clients. Web Proxy and firewall clients first attempt to communicate with the ISA Server to obtain configuration information. If the ISA Server does not respond, clients can direct WPAD or PSPAD queries to DNS or DHCP. See the section “Configuring Automatic Server Discovery” for more information.

Array Properties: The Performance Tab

The **Performance** tab has only one setting: a slider that tunes the servers in the array based on the number of users that are expected. Although the setting is configured by a slider, selections are not made on a continuous scale. You must select one of the following three labeled settings that specify the number of users (actually, connections) that are anticipated per day:

- **Fewer than 100**
- **Fewer than 1000**
- **More than 1000**

Ideally, the selection should be slightly higher than the actual number of connections experienced by ISA Server, a value that produces good performance without wasting server resources. If the setting is too low, ISA Server must allocate connection resources for each connection over the configured value, which can markedly slow the operation of a busy ISA Server.

This setting allocates sets of the resources that are required to support a connection. Rather than thinking of connections per day, think of simultaneous connections.

If ISA Server exhausts its pool of connection resources, additional resources can be allocated on a per-connection basis. Resource allocation takes time, and server responsiveness will be markedly slowed if large numbers of connections are configured *ad hoc*. If an ISA Server frequently runs out of pre-allocated connections, increase the setting on the **Performance** tab and, if necessary, add memory.

ISA Server includes several Performance Monitor counters that can be used to monitor ISA Server operations. Included are counters for numbers of TCP and UDP connections. These counters can be used to assist you in tuning the performance of the ISA Server.

Array Properties: The Security Tab

Security tabs are pretty familiar by now. Use this tab to specify users who can manage and use the ISA Server. By default, members of Domain Admins and Enterprise Admins have Full Control permissions. Authenticated Users have Read permissions.

Defining Policy Elements

Policy elements are groups of properties that can be referenced by a collective name from rules and filters. In most cases, policy elements are the only tools available for including multiple values in a field. As explained earlier in the section “Policy Elements,” there are seven types of policy elements. All types can be defined for arrays. Only five types can be defined for enterprises.

A given policy element can be used in any number of rules or filters. The advantage of this is that by editing a single policy element, you change the corresponding properties anywhere that policy element is used. It is often useful to create a policy element with a single entry just to take advantage of this labor-saving device.

We will look at the policy element dialog boxes in the next seven sections. Every dialog box has the following fields, which we’ll dispense with once and for all:

- The **Name** field identifies the policy element and is used in rules to reference the policy element.
- The **Description** field is optional. If the **Name** field does not fully explain the purpose of the policy element, enter additional descriptive material here.

In most cases, policy elements are managed using the following procedures.

To start creating a new policy element, do one of the following:

- If the right-hand pane does not include task icons, as in Details view and in some Taskpads, right-click the policy element type container in the object tree and choose **New**→*protocol element type* in the context menu.
- In Taskpad mode, if the Taskpad includes task icons, click the **Create protocol element type** button.

To modify an existing protocol element, select it in the list appearing in the Taskpad and do one of the following:

- If the right-hand pane does not include task icons, as in Details view and in some Taskpads, right-click the policy element and choose **Properties** from the context menu.
- If the Taskpad for a protocol element type includes task icons, select the protocol element and click the **Modify protocol element type** button.

To delete an existing protocol element, do one of the following:

- If the Taskpad does not include task icons, as in Details view and in some Taskpads, right-click the policy element, press the Delete key, and confirm your request.
- If the Taskpad for a policy element type includes task icons, select the policy element and click the **Delete protocol element type** button, and confirm your request.

Bandwidth Priorities

Bandwidth priorities policy elements are defined at the array level only. They are used in bandwidth rules to allocate WAN bandwidth to specific protocols. The **New Bandwidth Priority** dialog box is shown in Figure 34.

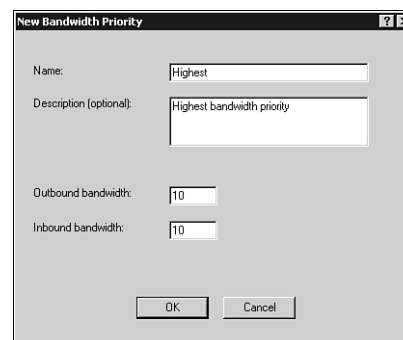


FIGURE 34

Policy Elements: Defining a Bandwidth Priorities policy element.

The **Outbound bandwidth** and **Inbound bandwidth** fields accept an integer from 0 through 200, where lower values represent higher priorities.

The effect of these parameters is ordinal, not relative. For example, 100 does not have twice the priority of 200. If several protocols are competing for bandwidth, the protocol with the lowest bandwidth value has the highest priority, the protocol with the next higher bandwidth value is next in priority, and the protocol with the highest bandwidth value has the lowest priority. The sequence 1-2-3-4-5 is functionally equivalent to the sequence 1-12-45-102-176. In practice, it is best to leave gaps in the numbering sequence to allow room to insert new bandwidth priorities in the sequence. A good approach is to number in intervals of 10: 10-20-30-40-50. This leaves room for a new highest or lowest priority as well as any priority in between.

Table 3 illustrates one way you might define a set of policy elements.

TABLE 3 Examples of Bandwidth Priority Policy Elements

<i>Name</i>	<i>Outbound bandwidth</i>	<i>Inbound bandwidth</i>
Highest	10	10
Medium High	20	20
Medium	30	30
Medium Low	40	40
Lowest	50	50
Highest Outbound	10	30
Highest Inbound	30	10

Client Address Sets

Client address sets are simply groups of computers, identified by their IP addresses. Client address sets can be used in several places to identify groups of computers.

Figure 35 shows the **Client Set** dialog box that is used to create and modify client address sets.

Click **Add** to open the **Add/Edit IP Addresses** list and define a range of IP addresses.

To modify an entry, select the entry and click **Edit**.

To delete an entry, select the entry and click **Remove**.

Content Groups

Content groups policy elements define the content types that are listed in site and content rules.

Figure 36 shows the ISA Server console displaying the initial set of content groups.

A content type is a MIME type that is defined by three properties:

- **A MIME content group.** ISA Server contains content types from eleven MIME content groups: application, application data files, audio, compressed files, documents, HTML documents, images, macro documents, text, video, and VRML.
- **A content type.** Typically, the content type identifies the application or protocol that creates the content. The full name of a MIME type consists of the MIME content group, associated with a content type by a slash, for example: application/msword. A list of content types that are predefined for ISA Server is included in ISA Server Help under the heading “Configure content groups.”
- **Filename extensions.** A MIME content type is associated with files by specific filename extensions, such as .doc for Microsoft Word document files.

When incoming content is transported by HTTP, the type is identified by the MIME type. When the MIME type is missing, or when incoming content is transported by FTP, the content type is identified by the filename extension.

To define a new content group right-click **Content Group** and select **New**→**Content Group** from the context menu. The **New Content Group** dialog box is shown in Figure 37.

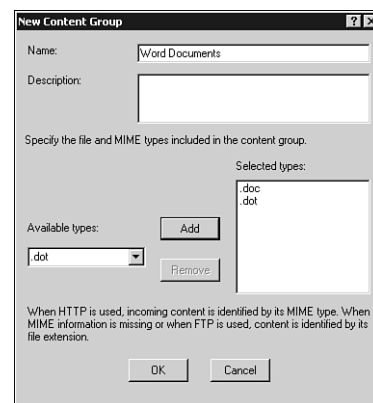


FIGURE 37

Policy Elements: Defining a Content Group Set policy element.

The content group consists of MIME types that are added to the **Selected types** list. Predefined MIME types and filename extensions can be selected from the **Available types** scroll box, with MIME types appearing first and the list and filename extensions after. Some MIME types include wildcards, for example, audio/* . These MIME type specifications include all MIME types in the specified content group.

Use the **Add** and **Remove** buttons to specify the entries in the **Selected types** list.

You can define additional MIME types and filename extensions by entering them in the **Available types** field and adding them to the **Selected types** list.

Destination Sets

Destination sets list groups of servers by address in the same manner that client address sets define sets of clients. Both types of policy elements are created and managed in the same manner.

Dial-up Entries

A dial-up entries policy element groups a dial-up connection with the user name and password credentials that gain access at the remote location. Dial-up entries policy elements can be used to automate the establishment of dial-up connections.

To create a new dial-up entries policy element, right-click **Dial-up entries** and choose **New**→**Dial-up Entry** from the context menu. Figure 38 shows the **New Dial-up Entry** form. Complete the form as follows:

- **Use the following network dial-up connection.** Enter the name of a dial-up connection that has been defined in **Network and Dial-Up Connections**. Click **Select** to choose from a list of existing dial-up connections.
- **User name.** Enter the user name for a user account that is authorized to connect through the chosen dial-up connection.
- **Password.** Enter the password for the user account.

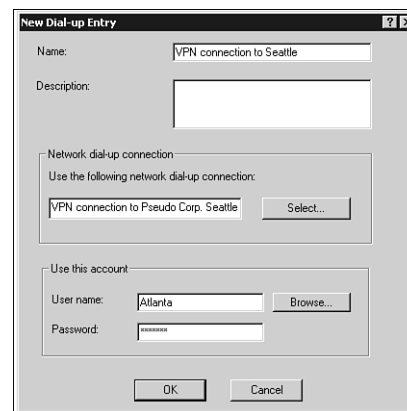


FIGURE 38

Policy Elements: Defining a Dial-Up Entries policy element.

If any dial-up entries policies are defined, one of them is always designated as the *active dial-up connection*. Some ISA Server capabilities, such as Web routing, use the active dial-up connection when dial-up connections are enabled. To select the active dial-up connection, do the following:

1. If Details view is not active, select **View**→**Taskpad** to remove the check mark.
2. In the Details pane, right-click the dial-up entries policy element that you want to make active and choose **Set as Active Entry**.

Protocol Definitions

Protocol definitions policy elements are used in protocol rules and packet filters to describe protocols that are allowed or blocked. Figure 39 shows the list of defined protocols in Details display mode. Quite a few protocol definitions are included with ISA Server, but it is often necessary to add protocols to the list.

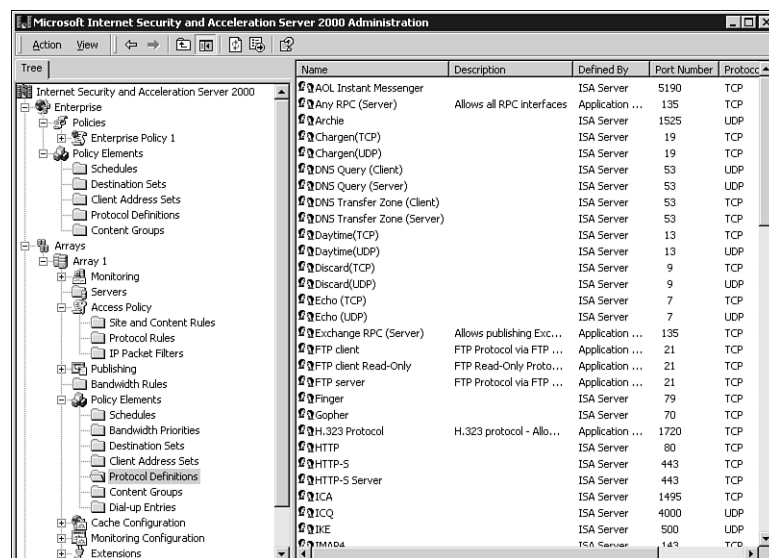


FIGURE 39

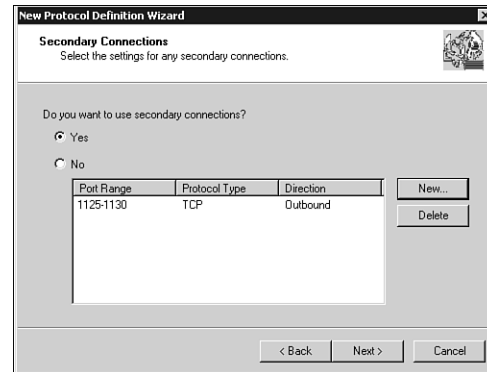
Some of the protocol definitions that are included with ISA Server.

Protocol definitions policy elements are created using the **New Protocol Definition Wizard**. Complete the wizard as follows:

1. In the first dialog box, enter a name for the protocol definition.
2. The **Primary Connection Information** dialog has three fields:
 - **Port number.** Enter the port number for the protocol.
 - **Protocol type.** Select UDP or TCP. Protocol definitions cannot be created for other host-to-host layer protocols, which must be managed using packet filters.
 - **Direction.** Select Outbound or Inbound. Unlike packet filters, protocol definitions apply to traffic in one direction only. Two protocol definitions must be created if it is necessary to apply rules to traffic in both directions.
3. The next dialog box, titled **Secondary Connections**, is shown in Figure 40. Some protocols need to open additional connections after the initial connection is established. These are known as *secondary connections* and are defined by a range of port numbers, a protocol type, and a direction. If secondary connections are required, do the following:
 - a. Select **Yes**. Active secondary connections are listed in the list box.
 - b. Click **New** to open the **New/Edit Secondary Connection** dialog box. Complete it as follows:
 - **From.** Enter the first port in the port range.
 - **To.** Enter the last port in the port range. No port in the range should conflict with other ports in use on the computer.
 - **Protocol type.** Select UDP or TCP.
 - **Direction.** Select Outbound or Inbound.
 - c. Click **OK** to add the secondary connection definition to the list.
4. Click **Finish** to complete the wizard and create the protocol definition.

NOTE

The built-in protocol definitions often identify definitions for inbound protocols by including the designation (*Server*) in the name, meaning that the anticipated communication is directed from a server that is outside the firewall to a client that is inside. Protocol definitions for outbound protocols often include the designation (*Client*) indicated communication directed from an inside client to an outside server. There is nothing inherent in a (*Server*) protocol definition that requires packets to be originated by a server, just as a (*Client*) protocol definition does not require that packets be originated by a client. The labels only indicate the direction of the packet's travel: outside-in or inside-out.

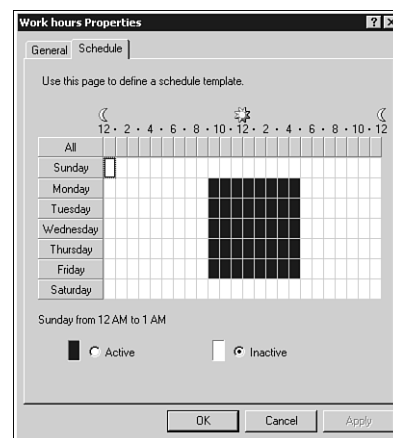
**FIGURE 40**

Defining secondary connections for a protocol definition.

Schedules

Schedules policy elements define blocks of time. They provide a convenient method of defining time schedules for inclusion in rules.

The schedule property of a schedules policy element is defined using the blue-and-white grid we have seen several times before in this book. Figure 41 shows the schedule grid for a schedules policy element named **Work hours**. The schedule is active during the time blocks that are colored blue.

**FIGURE 41**

Policy Elements: Defining a Schedules policy element.

ISA Server includes two pre-configured schedules that can be modified: **Work hours** mentioned above and **Weekends**, which is active on Saturday and Sunday from 12 A.M. to 12 A.M.

A schedule named **Always** is always available and can't be modified. The **Always** schedule is active 24 hours a day, 7 days a week.

Defining Protocol Rules

When it is initially installed, ISA Server blocks all outbound and inbound packets, simply because no rules or protocol filters exist. In most cases, you will want to enable outbound communication by defining one or more protocol rules and at least one site and content rule to enable the Web Proxy service to process Web requests.

Defining a Protocol Rule

Two icons on the **Protocol Rule** Taskpad initiate the **New Protocol Rule Wizard**. We'll start with the **Create Protocol Rule** icon. In the next section we will identify the differences that take place when the wizard is started from the **Allow Web protocols** icon.

To create a protocol rule:

1. Select **Protocol Rules** in the object tree and start the **New Protocol Rule Wizard** by doing one of the following:
 - In Details view, right-click **Protocol Rules** and choose **New→Rule** from the context menu.
 - In Taskpad view, click the **Create Protocol Rule** icon.
2. In the **Welcome** dialog box, enter a descriptive name for the protocol rule.
3. In the **Rule Action** dialog box, select **Allow** or **Deny** to specify how ISA Server will respond when a client attempts to communicate with protocols covered by the rule.
4. The next dialog box is **Protocols**. Select one of the following choices in the drop-down menu:
 - **All IP traffic**. Select this option if the protocol applies to all packets with an IP payload. This rule can still be overridden by a rule or packet filter that denies an IP protocol, or by a site and content rule that blocks access for specific clients or for specific time blocks.
 - **Selected protocols**. When this option is selected, the **Protocols** list box is displayed, along with the **Select All** and **Clear All** buttons and the **Show only selected protocols** check box. Figure 42 shows the appearance of the **Protocols**

dialog when **Selected protocols** is chosen. Procedures for completing the **Protocols** list are discussed in Step 5.

- **All IP Traffic Except Selected**. This option also produces a **Protocols** dialog box like the one in Figure 42. This time, however, ISA Server blocks the protocols that are checked.

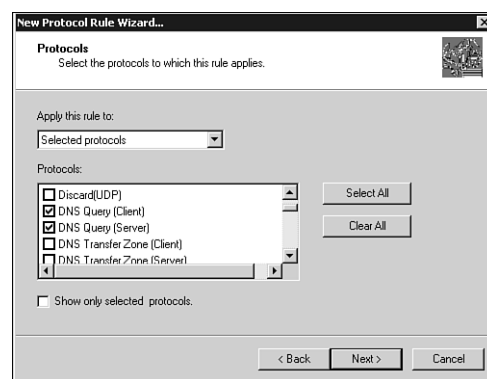


FIGURE 42

Protocol Rules: Specifying the protocols.

5. Protocols in the **Protocols** list are selected, logically enough, by checking protocols. The manner in which the check marks are interpreted is determined by the selection under **Apply this rule to**. If **Selected protocols** is chosen, check boxes to allow the protocol. If **All IP Traffic Except Selected** is chosen, check protocols to deny them.

The **Show only selected protocols** check box determines whether all protocols or selected protocols will be displayed in the **Protocols** list. Check this box to display the selected protocols in a compact form. Remove the check to select new protocols. Protocols appear in the **Protocols** list only if they are defined as protocol definitions policy elements.

Select the desired protocols and move on.

6. After the **Protocols** dialog box comes **Schedule**. Select a schedule from the list, which includes the **Always** schedule as well as schedules defined in schedules policy elements.
7. Next is the **Client Type** dialog box, which specifies the clients that are affected by the protocol rule. There are three choices:
 - **Any user, group, or client computer**. This selection applies the rule to all local clients. This is often what you want, because often all clients have the same Web access privileges.

- **Specific computer (client address sets).** When this button is selected, the next dialog box is **Client Sets**. There you must select client address sets policy elements that describe the desired clients.
 - **Specific users and groups.** If this button is selected, the next dialog box is **Users and Groups**. There you can select user accounts and groups taken from Active Directory or, in the case of a server that does not belong to a domain, the ISA Server's local account database. This method of identifying clients will work only for Windows clients that are members of the same domain as the ISA Server. To define rules for non-Windows computers, select the previous option and identify clients by their IP addresses.
8. Finish the wizard to create the protocol rule.

All properties in a protocol rule can be modified after the rule is created.

NOTE

Protocol rules and site and content rules are not processed in any particular order, but ISA Server does process deny rules before it processes allow rules. Processing stops when ISA Server identifies a rule that applies to the request. Consequently, if a deny rule is encountered, ISA Server denies access without processing any allow rules.

NOTE

Rules can be disabled without deleting them. To disable a rule, do the following:

1. If the console is not in Details view, select **View→Taskpad**.
2. In the Details pane select the container that holds the rule to be disabled.
3. Right-click the rule in the Details pane and choose **Disable** in the context menu.

A disabled rule is marked with a red, downward-pointing arrow beside its icon in the Details pane.

Repeat the procedure, choosing **Enable** in Step 3, to enable a disabled rule.

Defining a Protocol Rule for Web Protocols

The procedure described in this section can be performed only in Taskpad view.

Nearly all ISA Servers will be required to support outgoing Web requests, and there is a Taskpad icon dedicated to creating a protocol rule that enables common Web protocols. The **Protocol Rules** Taskpad includes a button labeled **Allow Web protocols**. This button starts the **New Protocol Rule Wizard** but pre-configures several properties. The properties and values that are preset are

1. The **Rule Action** dialog is not displayed. The rule action is preset to **Allow**. The setting cannot be changed in the wizard, but it can be modified when the protocol is edited after the wizard is completed.
2. The **Protocols** dialog box is initialized with these settings:
 - **Apply this rule to** is preset to **Selected protocols**.
 - Protocols **FTP client**, **FTP client Read-Only**, **Gopher**, **HTTP**, and **HTTPS** are pre-selected.
3. **Always** is pre-selected in the **Schedule** dialog box.
4. **Any user, group, or client computer** is pre-selected in the **Client Type** dialog box.

The **Allow Web protocols** icon is a convenience, but it isn't essential. Everything that **Allow Web protocols** accomplishes can be configured nearly as easily by using the **New Protocol Rule Wizard** without the Web-based presets.

Defining Site and Content Rules

Site and content rules allow or deny access to Web servers via the Web Proxy, but do not allow or deny use of protocols. They can't enable a client to use a protocol that is not allowed by a protocol rule. Before creating site and content rules, you need to define one or more protocol rules that enable clients to use Web protocols, perhaps by defining a protocol rule with the procedure described in the previous section.

After a Web protocol rule is created, define a site and content rule as follows:

1. Start the **New Site and Content Rule Wizard** using one of the following methods.
 - In **Details** view, right-click **Site and Content Rules** and choose **New→Rule** from the context menu.
 - In **Taskpad** view, select **Site and Content Rules** in the object tree. Then click the **Create site and content rule** icon.
2. In the **Welcome** dialog box, enter a name for the rule.

3. Complete the **Rule Action** dialog box (Figure 43) as follows:

- Select **Allow** to permit users affected by the rule to access Web services on the requested Web server.
- Select **Deny** to deny Web access to users affected by the rule or to redirect requests to another Web server.
- **If HTTP request, redirect request to this site.** This check box is active only if **Deny** is selected. Check it if HTTP requests that are denied should be redirected to another Web server. Then enter the URL for the Web server that will receive the redirected requests.



FIGURE 43

Site and Content Rules: Specifying the rule action.

4. Next is the **Destination Sets** dialog box. Under **Apply this rule to**, select one of the following options:

- **All destinations.** The rule applies to all destination computers, internal and external.
- **All internal destinations.** The rule applies only when a service request is sent to an internal Web server.
- **All external destinations.** The rule applies only when a service request is sent to an external Web server.
- **Specified destination set.** The rule applies only to members of a specific destination sets policy element, which are discussed in the section “Destination Sets” earlier in the chapter. When this option is selected, a drop-down box is displayed from which you can select any existing destination set.
- **All destinations except selected set.** The rule applies to all destinations except members of a specific destination sets policy element. When this option is selected, a drop-down box is displayed from which you can select any existing destination set.

5. Next, the **Schedule** dialog box appears with the **Use this schedule** field pre-configured with **Always**. To change the schedule, pull down the list box and select an existing schedule policy element. The list includes the built-in schedules **Always**, **Weekends**, and **Work hours**, as well as any schedules policy elements administrators have defined.
6. Next is the **Client Type** dialog box, which specifies the clients that are affected by the protocol rule. Available choices are
 - **Any user, group, or client computer**
 - **Specific computer (client address sets)**
 - **Specific users and groups**These choices are discussed in the section “Defining Protocol Rules.” See step 7 of the procedure in that section.
7. Finish the wizard to create the site and content rule.

Defining Web Publishing Rules

Web publishing rules enable outside clients to communicate with internal Web servers. Web publishing rules function differently from protocol rules and site and content rules in that Web publishing rules are processed in a specific order. Processing ends when a rule is found that matches the conditions of the Web service request.

The list of Web publishing rules ends with a pre-defined rule titled **Last**, which is always at the bottom of the list. Rules created by an administrator can be repositioned in the processing order, but the **Last** rule is always the final rule in the list. The **Last** rule denies everyone access to anything at any time. Thus, the default state of ISA Server prohibits all outside access to inside Web servers.

To create a new Web publishing rule:

1. Start the **New Web Publishing Rule Wizard** using one of the following methods:
 - In Details view, right-click **Arrays**→*array*→**Publishing**→**Web Publishing Rules** in the object tree. Then choose **New**→**Rule** in the context menu to start the **New Web Publishing Rule Wizard**.
 - In Taskpad view, select **Arrays**→*array*→**Publishing**→**Web Publishing Rules** in the object tree. Then click the **Create Web Publishing Rule** icon to start the wizard.
2. In the **Welcome** dialog box, enter a descriptive name for the Web publishing rule.

3. In the **Destination Sets** dialog box under **Apply this rule to**, select one of the following options:

- **All destinations**
- **All internal destinations**
- **All external destinations**
- **Specified destination set**
- **All destinations except selected set**

These options are explained in the section “Defining Site and Content Rules.”

4. Next is the **Client Type** dialog box, which specifies the clients that are affected by the protocol rule. There are three choices:

- **Any user, group, or client computer**
- **Specific computer (client address sets)**
- **Specific users and groups**

These choices are described in the section “Defining Protocol Rules.”

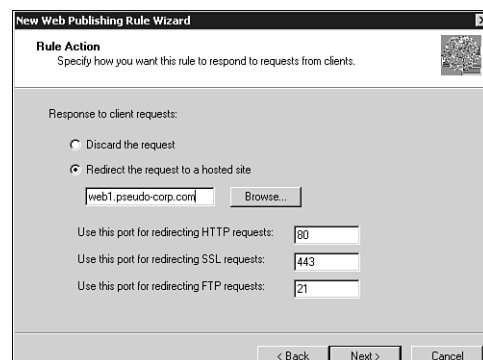
5. The next dialog box is **Rule Action**, shown in Figure 44. There are two primary options:

- **Discard the request.** Rules configured with this option block incoming requests that match rule conditions.
- **Redirect the request to a hosted site.** Rules configured with this option forward incoming requests to an internal Web server. (Incoming requests that are not discarded must always be redirected because the URL identifies ISA Server as the target server. ISA Server must map that public destination to a destination on the private network.)

When the rule instructs ISA Server to redirect requests, the destination must be defined.

These fields define the target Web server:

- In the field following **Redirect the request to a hosted site**, identify the destination in one of these ways: IP address, domain name, or Active Directory computer name. Click **Browse** to select a computer name from Active Directory.
- **Use this port for redirecting HTTP requests.** In most cases, Web servers receive HTTP requests on port 80, which is the default value for this field. There are times, however, when a different port must be specified. See “Co-Hosting IIS on a computer with ISA Server” for one example.
- **Use this port for redirecting SSL requests.** The default SSL port is 443. This port seldom changes, but a different value can be entered if necessary.
- **Use this port for redirecting FTP requests.** The default FTP control port is 21. This port seldom changes, but a different value can be entered if necessary.

**FIGURE 44**

Web Publishing Rules: Specifying the rule action.

6. Finish the wizard to create the Web publishing rule.
7. Next, you need to place the new rule in the desired position in the Web publishing rule list. Adjust the processing order of the new Web publishing rule by doing the following:
 - a. If the console is not in Details view, choose **View**→**Taskpad**. (This procedure cannot be performed in Taskpad view.)
 - b. In the Details pane, right-click the rule to be moved and choose **Move Down** or **Move Up** as is appropriate.

Defining Server Publishing Rules

Server publishing rules are used to publish non-Web services. They could be used for Web protocols, but there would be no support for caching or forwarding of requests. A server publishing rule simply opens up a channel that enables an incoming service request to be directed to a particular server.

No server publishing rules are pre-defined. There is no default rule and server publishing rules are not processed in a particular order. Server publishing rules always allow communication with the target service and must be disabled or deleted to eliminate their effects.

To create a server publishing rule, proceed as follows:

1. Start the **New Web Publishing Rule Wizard** using one of the following methods:
 - In Details view, right-click **Arrays**→*array*→**Publishing**→**Server Publishing Rules** in the object tree. Then choose **New**→**Rule** in the context menu.
 - In Taskpad view, select **Arrays**→*array*→**Publishing**→**Server Publishing Rules** in the object tree. Then click the **Publish Server** icon to start the **New Server Publishing Rule Wizard**.

2. In the **Welcome** dialog box, enter a descriptive name for the Web publishing rule.
3. Complete the following fields in the **Address Mapping** dialog box:
 - **IP address of internal server**
 - **External IP address on ISA Server**

The ISA Server external interface may have a single IP address or it may be multi-homed. If you are publishing several internal servers for a service and want to use standard port numbers on the external interface, you must configure the external ISA Server interface with a separate IP address for each internal server.
4. In the **Protocol Settings** dialog box, select the protocol to be published. The supported protocols are described earlier in the section “Server Publishing Rules.”
5. Next is the **Destination Sets** dialog box. Under **Apply this rule to**, select one of the following options:
 - **All destinations**
 - **All internal destinations**
 - **All external destinations**
 - **Specified destination set**
 - **All destinations except selected set**

These options are explained in the section “Defining Site and Content Rules.”
6. Next is the **Client Type** dialog box, which specifies the clients that are affected by the protocol rule. There are two choices:
 - **Any user, group, or client computer**
 - **Specific computers (client address sets)**

These choices are described in the section “Defining Protocol Rules.”
7. If **Specific computers (client address sets)** is chosen in the **Client Type** dialog box, the next dialog box is **Client Sets**. Add one or more pre-defined client address sets policy elements to the list.
8. Finish the wizard to create the server publishing rule.

Because you have no control over the order in which server publishing rules are processed, you must take care to ensure that only one rule applies to a given set of circumstances. Suppose you have created two rules named **POP3 mail** and **POP3 server** with the following properties:

- **External IP address of ISA Server:** 209.51.67.5
- **Protocol Settings:** POP3 (Server)
- **Client type:** Any user, group, or client computer

The rule **POP3 mail** specifies **IP address of internal server** as `10.1.0.25`.

The rule **POP3 mail** specifies **IP address of internal server** as `10.1.0.30`.

When ISA Server receives POP3 requests on address `209.51.67.5`, where will it redirect the request? It could be either `10.1.0.25` or `10.1.0.30`. Apart from eliminating the rule conflict, you have no way of being certain that the desired server will receive the redirected requests.

Fortunately, in Details view all the critical properties of server publishing properties can be displayed, making it easy to scan the properties of all existing rules and spot conflicts. A good way to start is to sort by protocol and ensure that no rules conflict for the same protocol.

Configuring Web Proxy Routing

Routing rules determine how the Web Proxy service forwards outbound and inbound Web service requests. Like Web publishing rules, routing rules are processed in the order specified by the administrator. A default routing rule is always positioned at the end of the list. The default routing rule specifies that all requests are to be retrieved directly from the destination Web server. Rules you create will always have a higher priority than the default routing rule.

For incoming Web service requests, ISA server first processes Web publishing rules. Then it processes routing rules. The division of labor is as follows:

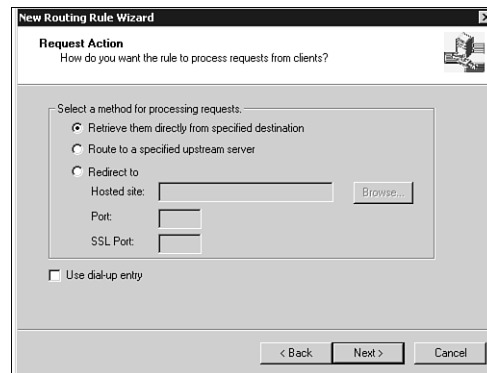
1. The Web publishing rule either drops the request or redirects it to an internal Web server.
2. A routing rule determines how to reach the Web server specified by the Web publishing rule. If the specified destination cannot be matched to a routing rule, the request is dropped.

To create a new routing rule:

1. Start the **New Routing Rule Wizard** by right-clicking **Arrays**→*array*→**Publishing**→**Web Publishing Rules** in the object tree and choosing **New**→**Rule** in the context menu. (RC1 does not have a Taskpad icon for this function.)
2. In the **Welcome** dialog box, enter a descriptive name for the routing rule.
3. Next is the **Destination Sets** dialog box. Under **Apply this rule to**, select one of the following options:
 - **All destinations**
 - **All internal destinations**
 - **All external destinations**
 - **Specified destination set**
 - **All destinations except selected set**

These options are explained in the section “Defining Site and Content Rules.”

4. The next dialog box is **Request Action**, which is shown in Figure 45. Options in this dialog box determine how the Web Proxy will process outgoing requests. The following options are available:
- **Retrieve them directly from specified destination.** The Web Proxy will send the query to the requested destination on behalf of the client.
 - **Route to a specified upstream server.** This option is used when ISA Servers are chained. When it is selected, the next dialog box is **Primary Routing**. Use this option for routing outbound requests only.
 - **Redirect to.** When this option is selected, the Web Proxy directs the request to a server that is different from the one specified by the client. To configure redirection, you must complete the following fields:
 - **Hosted site.** Enter the IP address, domain name, or Active Directory name of the Web server. Click **Browse** to browse Active Directory for a name.
 - **Port.** Specify the HTTP port on the target server.
 - **SSL Port.** Specify the HTTPS port on the target server.
 - **Use dial-up entry.** This check box is active only if at least one dial-up entries policy element is defined and either **Retrieve them directly from specified destination** or **Route to a specified upstream server** is selected. No dial-up entry is specified in the routing rule. The active dial-up connection is used. See the section “Dial-up Entries” for information about the active dial-up connection.

**FIGURE 45**

Routing Rules: Specifying the Web Proxy request action.

CAUTION

Route to a specified upstream server should never be selected for routing rules that apply to inbound Web requests because it always routes the request to an upstream ISA Server; that is to say, an ISA Server that is outside the boundaries of the network to which the target server is connected. The Web service request will never be routed to the desired internal Web server.

5. If **Route to a specified upstream server** is selected in the **Request Action** dialog box, the next dialog box is **Primary Routing**, shown in Figure 46. Complete the dialog box as follows:
 - **Server or array.** Enter the IP address, domain name, or Active Directory name of an upstream server, or enter the array name of an upstream array.
 - **Port.** Enter the port on which the upstream server or array receives outgoing HTTP requests. The default port is 8080, which is the default HTTP port for Web Proxy clients.
 - **SSL Port.** Enter the port on which the upstream server or array receives outgoing HTTPS requests. The default port is 8443, which is the default HTTPS port for Web Proxy clients.
 - **Use this account.** Check this box if the upstream server or array requires authentication. Then enter the authentication information as follows:
 - **User name.** Enter a user name that is accepted by the upstream server.
 - **Password.** Enter the password for the user account.
 - **Authentication.** Select **Basic** for clear text authentication. Select **Integrated Windows** to use secure, Windows-based authentication.
6. If **Route to a specified upstream server** is selected in the **Request Action** dialog box, the next dialog box is **Backup Routing**. This option defines a backup route that is used if the primary route is unavailable for any reason. Select one of the following options:
 - **Ignore requests.** Requests are dropped if the primary route fails.
 - **Receive requests directly to specified location.** If the primary route fails, the Web Proxy bypasses the upstream server and sends the request directly to the Web server that is specified in the original request. Obviously, for this option to work there must be an alternate route to the Web server that is not affected by a failure in the private route. When this choice is selected, the **Use dial-up entry** option is active and can be checked to enable use of the active dial-up connection to establish the secondary route.

- **Route requests to an upstream server.** If the primary route fails, the Web Proxy service sends the request to a backup ISA Server or array. When this choice is selected, the **Use dial-up entry** option is active and can be checked to enable use of the active dial-up connection.
7. If **Route requests to an upstream server** is selected in the **Backup Routing** dialog box, the next dialog box requests routing and authentication information for the backup upstream server or array. This dialog box has the same options as the **Primary Routing** dialog box shown in Figure 46.
 8. The next dialog box is **Cache Configuration** (See Figure 47), which specifies how the Web Proxy service should search for an object.

The screenshot shows the 'New Routing Rule Wizard' dialog box, specifically the 'Primary Routing' step. The title bar reads 'New Routing Rule Wizard'. Below the title, the section is titled 'Primary Routing' with a subtitle: 'You can specify the primary route for requests that are sent to an upstream proxy server or array.' The main instruction says: 'Type the name of the server and the port number for the primary route. If you want to use a specific user account, type the name and password.' The form contains the following fields and options:

- 'Server or array:' text box containing 'Array 3' and a 'Browse...' button.
- 'Port:' text box containing '8080'.
- 'SSL Port:' text box containing '8443'.
- A checked checkbox labeled 'Use this account:'.
- 'User name:' text box containing 'Jimmy Thudpucker' and a 'Browse...' button.
- 'Password:' text box with masked characters.
- 'Authentication:' dropdown menu set to 'Integrated Windows'.

At the bottom of the dialog are three buttons: '< Back', 'Next >', and 'Cancel'.

FIGURE 46

Routing Rules: Defining the primary route.

The screenshot shows the 'New Routing Rule Wizard' dialog box, specifically the 'Cache Configuration' step. The title bar reads 'New Routing Rule Wizard'. Below the title, the section is titled 'Cache Configuration' with a subtitle: 'Specify how the rule searches the cache for requested objects and how to route requests if no valid object exists.' The main instruction says: 'Search cache for:'. The form contains the following options:

- A radio button selected for the option: 'A valid version of the object; if none exists, retrieve the request using the specified requested action'.
- An unselected radio button for the option: 'Any version of the object; if none exists, retrieve the request using the specified request action'.
- An unselected radio button for the option: 'Any version of the requested object. Never route the request'.
- An unselected checkbox for the option: 'Never cache the response to the request'.

At the bottom of the dialog are three buttons: '< Back', 'Next >', and 'Cancel'.

FIGURE 47

Routing Rules: Configuring the cache.

Select one of the following options:

- **A valid version of the object; if none exists, retrieve the request using the specified requested action.** The Web Proxy service will search its local cache, or its array cache, for a valid version of the target object. A version is considered valid if its TTL has not expired. If the object is not available in cache, the request is sent to the server specified in the request. This option is appropriate in most circumstances.
- **Any version of the object; if none exists, retrieve the request using the specified request action.** The Web Proxy service will search its local cache, or its array cache, for any available version of the target object, even a version whose TTL has expired. If the object is not available in cache, the request is sent to the server specified in the request.

CAUTION

Be wary of the **Any version of the object** option. TTL settings specify the lifetimes of cached objects. The second option essentially ignores TTL and could result in ISA Server returning old objects to clients when a more recent version can be found on the Web server.

- **Any version of the requested object. Never route the request.** The Web Proxy service will search its local cache, or its array cache, for any available version of the target object, even a version whose TTL has expired. If the object is not available in cache, the cache does not search for the object on a Web server.

CAUTION

This option should be used with caution because it makes no provision either for storing objects in cache initially or for resolving the request if the desired object is not in cache. With this option, the administrator must ensure that desired objects are loaded into cache. One useful technique for loading the cache is to employ scheduled content download jobs, scheduled to execute periodically. See the section "Scheduling Content Downloads" for additional information.

Also present is the **Never cache the response to the request** check box. If the option is selected, ISA Server will not cache the results of queries from clients affected by this Web publishing rule.

NOTE

Should you cache or not? In most cases, caching is appropriate. The TTL provided by the Web server enables old objects to be removed from cache. But there are many instances when caching should be disabled. A Web routing rule for a Web server that generates dynamically generated pages or communicates entirely using secure HTTP should disable caching in most cases. In the first case, there is no point in caching the objects because they are valid only for a particular user, query, or time. In the second case, Web responses that use SSL should not be cached because they can be opened only by the owner and, because they are time-stamped, can be opened only for a short time after they are generated.

9. Finish the wizard to create the routing rule.
10. Routing rules are processed according to the order in which they appear in the rule list. Adjust the processing order of the new Web publishing rule by doing the following:
 - a. If the console is not in Details view, choose **View**→**Taskpad**.
 - b. In the Details pane, right-click the rule to be moved and choose **Move Down** or **Move Up** as is appropriate.

Defining IP Packet Filters

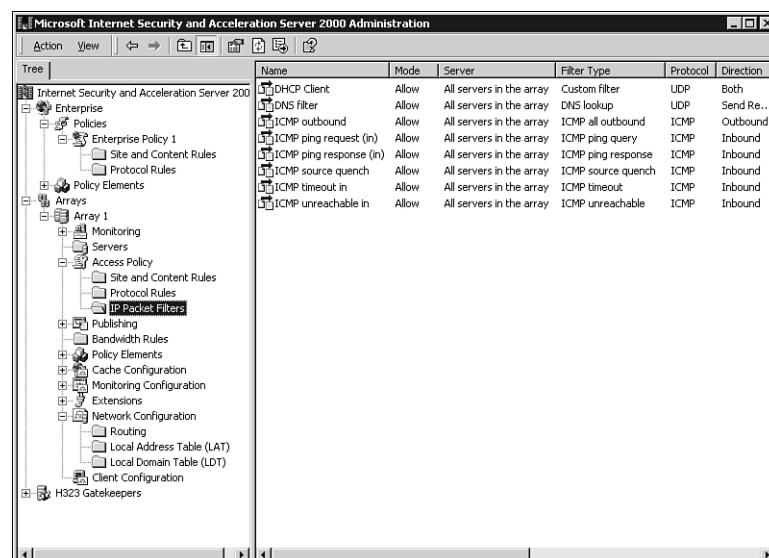
Often packet filters provide the only mechanism that can support a desired ISA Server function. A good example of a configuration that can be enabled only with packet filters is found in the later section “Co-Hosting IIS on a Computer with ISA Server.”

ISA Server is pre-configured with several packet filters. To display packet filters select **Arrays**→**array**→**Access Policy**→**IP Packet Filters**. Due to the number of columns, I find Details view the most useful method of viewing the list of packet filters. Figure 48 shows the packet filters list in the ISA Server console.

Packet filtering must be enabled before packet filters take effect, so before we look at the procedures for creating packet filters, we need to examine the procedures for activating packet filter support on arrays. Along the way, we will also encounter several options related to packet filtering.

Enabling and Configuring IP Packet Filtering

IP packet filtering can be enabled at the enterprise or array level. The section “Configuring Enterprise Policy Default Settings” explains how to force an array to use packet filtering.

**FIGURE 48**

Packet filters shown in Details view.

The **Policies** tab in the **Array Properties** dialog box can also enable packet filtering for a specific array. See the discussion in the section “Array Properties: The Policies Tab” for information on that approach.

The **IP Packet Filters Properties** dialog box, in addition to enabling packet filtering, is used to configure several packet filtering options that aren’t addressed anywhere else. To open this dialog box do one of the following:

- Right-click **IP Packet Filters** and choose **Properties** from the context menu.
- In Taskpad view, select **IP Packet Filters** and click the **Configure Packet Filtering and Intrusion Detection**.

IP Packet Filters Properties: The General Tab

The **General** tab has three options:

- **Enable packet filtering.** This, of course, forces the array to support IP packet filtering. This capability can be enabled or disabled at the enterprise or array level, in which case packet filtering cannot be enabled here. Alternatively, packet filtering may be allowed at the enterprise or array levels, in which case it can be enabled optionally here. The remaining options on this tab are active only if packet filtering is activated for the array.

- **Enable intrusion detection.** As discussed in the section “Intrusion Detection,” ISA Server can be figured to detect several types of potentially damaging packets and to notify administrators of the packets’ presence on the network. If this option is checked, settings on the **Intrusion Detection** dialog box can be enabled.
- **Enable IP routing.** Check this option to enable ISA Server to route packets.

CAUTION

Never enable IP routing without enabling IP packet filtering. The combination of active IP routing and inactive packet filtering configures the computer to route IP traffic without security controls, bypassing ISA Server entirely. That might be useful when trying to isolate the source of a communication problem, but it violates the entire purpose for installing ISA Server.

IP Packet Filters Properties: The Packet Filters Tab

The **Packet Filters** tab has three check boxes:

- **Enable filtering of IP fragments.** IP can fragment and reassemble large datagrams, although IP fragmentation is not as efficient as fragmentation performed by host-to-host protocols. In most cases TCP/IP protocol stacks are designed to fragment messages in TCP or UDP so that IP fragmentation won’t be required in normal communication. However, some intrusion techniques are associated with IP fragments. Enable this option if ISA Server should drop IP fragments. Be aware, however, that some legitimate processes on your network may generate IP fragments. Although legitimate IP fragmentation is rarely encountered, you may want to use Network Monitor to examine traffic on your network for a period of time to ensure that no crucial traffic incorporates IP fragments.
- **Enabling filtering IP options.** Like many protocols, several options can be appended to IP headers. In practice, however, IP options are seldom encountered. Some attacks involve use of IP options, and you may wish to enable this option to drop packets whose IP headers contain options.
- **Log packets from ‘Allow’ filters.** If this option is selected, ISA Server will generate a log entry for each packet that is passed to an Allow packet filter. On a busy ISA Server, such logging will generate large logs and will probably slow server performance. You will probably want to enable this option only when you are verifying the correct operation of filters.

IP Packet Filters Properties: The Intrusion Detection Tab

The **Intrusion Detection** tab is shown in Figure 49. Options on this tab are active only if **Enable intrusion detection** is checked on the **General** tab. The intrusion techniques that ISA Server can detect are discussed in the section “Intrusion Detection” earlier in the chapter. I leave it to you to use your wits and determine how intrusion detection is enabled for each type of attack.

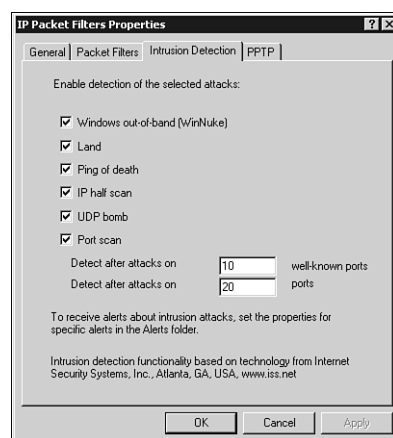


FIGURE 49

IP Packet Filters Properties: The Intrusion Detection tab.

As discussed earlier, port scans consist of efforts to scan ranges of ports to determine which ports are open on the computer and may, therefore, provide entry points for intrusion. When **Port scan** is enabled, two properties must be defined:

- **Detect after attacks on n well-known ports.** Recall that the well-known ports, numbered 0 through 1023, are assigned by IANA to particularly prominent Internet protocols. Because the protocol assignments for these ports do not change, when an intruder identifies an open port it is easy to identify the service that is running on that port. Good network intruders are very familiar with the inner workings of many of these protocols and can often exploit subtle vulnerabilities or newly discovered methods of hacking the protocol. Consequently, you may want to keep a close eye on these protocols by signaling a port scan on a fairly small number of well-known ports. The default number of ports is 10.
- **Detect after attacks on n ports.** This option addresses the entire range of ports from 0 through 65,535. Because there are so many ports, you probably want to raise the alarm threshold for general port scans to prevent false positive alerts.

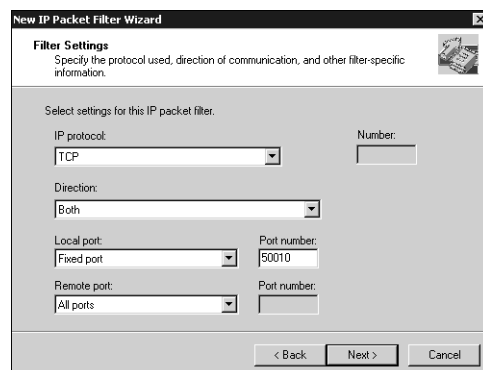
IP Packet Filters Properties: The PPTP Tab

By default, ISA Server does not channel VPN tunnels based on PPTP. To enable PPTP support, check **PPTP through ISA firewall**.

Creating IP Packet Filters

To create a packet filter, proceed as follows:

1. Start the **New IP Packet Filter Wizard** using one of the following methods:
 - Right-click **Arrays**→*array*→**Access policy**→**IP Packet Filters** in the object tree. Then choose **New**→**Filter** in the context menu.
 - In Taskpad view, select **Arrays**→*array*→**Access policy**→**IP Packet Filters** in the object tree. Then click the **Create Packet Filter** icon.
2. In the **Welcome** dialog box, enter a name for the packet filter.
3. In the **Servers** dialog box, select one of the following options to determine which computers are affected by this filter:
 - **All ISA Server computers in the array.**
 - **Only this server.** When you choose this option, select one of the servers in the array from the pull-down list.
4. In the **Filter Mode** field, specify the filter's behavior by selecting one of the following options:
 - **Allow packet transmission**
 - **Block packet transmission**
5. In the **Filter Type** dialog box, identify the protocol that is filtered by choosing one of the following:
 - **Custom.** If this option is selected, the next dialog box is **Filter Settings**, which enables you to describe a protocol that is not defined in a protocol definitions policy element.
 - **Predefined.** If this option is selected, choose one of the protocols in the pull-down list. The list will include all existing protocol definitions policy elements.
6. If **Custom** is selected in the **Filter Type** dialog box, the next dialog box is **Filter Settings**, shown in Figure 50. Here you can describe virtually any IP protocol. Complete the dialog box as follows:
 - **IP Protocol.** Select one of the following:
 - **Custom protocol.** If you make this choice, a protocol ID must be entered in the **Number** field.
 - **Any.** The rule applies to all IP protocols. The only option that can be configured when **Any** is selected is **Direction**.

**FIGURE 50**

Describing the type of an IP packet filter.

- **ICMP.** The filter can be made to apply to all ICMP message types or a specific ICMP message type.
- **TCP.** The filter can be made to apply to all TCP packets or to a specific TCP protocol.
- **UDP.** The filter can be made to apply to all UDP packets or to a specific UDP protocol.
- **Direction.** These three choices are available when **TCP**, **Any**, or **Custom protocol** is selected in the **IP Protocol** dialog box:
 - **Both** (inbound and outbound)
 - **Inbound**
 - **Outbound**

These five choices are available when **UDP** or **ICMP** is selected in the **IP Protocol** dialog box:

- **Receive only.** Only packets sent from an external computer to an internal computer.
- **Send only.** Only packets sent from an internal computer to an external computer.
- **Both.** Packets sent or received by an internal computer.
- **Receive send.** Packets sent from an external computer to an internal computer as well as packets the internal computer sends in response.
- **Send receive.** Packets sent from an internal computer to an external computer as well as packets the external computer sends in response. This option could be used to enable an inside computer to ping an outside computer and receive the outside computer's response.

- **Local port** and **Remote port.** These fields are displayed when TCP or UDP is selected in the **IP Protocol** field. For each field you can make one of the following selections:
 - **All ports.** The rule applies generally to all TCP or UDP traffic traveling in the specified direction from or to the port.
 - **Fixed port.** The rule applies to one port. The port number must be specified.
 - **Dynamic (1025-5000).** It often happens that a client requesting a service will direct the request to a fixed port on the server from a port on the client that is dynamically selected to not conflict with ports already in use. When this option is chosen, the port number cannot be entered because it is not specified until the request is made.
 - **Type** and **Code.** These fields are displayed when ICMP is selected in the **IP Protocol** field. For each field you can make one of the following selections:
 - **All Types** or **All Codes.** The rule applies generally to all ICMP traffic.
 - **Fixed Type** or **Fixed Code.** The rule applies to a specific ICMP type or code. The number associated with the type or code must be specified.
7. The next dialog box is **Local Computer**. These settings determine which local interfaces the packet filter applies to. (Remember that filters are applied to the external interface.) One of the following three choices must be selected:
- **Default IP addresses for each external interface on the ISA Server computer.** The filter applies to all IP addresses assigned to the server's external interface.
 - **This ISA server's external IP address.** An IP address belonging to an external interface of one of the servers in the array must be specified when this option is selected. The filter is applied to only that interface. This option is used to define filters for servers on the peripheral network when a three-homed architecture is deployed. (See "The Three-Homed Perimeter Network Configuration" earlier in this chapter.)
 - **This computer (on the perimeter network).** ISA Server applies this filter to packets traveling to or from (depending on the **Direction** selected in the **Filter Settings** dialog box).
8. The next dialog box is **Remote Computer**. These settings determine which interfaces on the remote computer the packet filter applies to. There are two choices:
- **All remote computers.** This selection might be used to enable one or more local computers to access all external servers that provide a service.
 - **Only one remote computer.** If this option is selected, an IP address must be entered to identify the remote computer.
9. Finish the wizard to create the filter.

Modifying IP Packet Filters

Many IP packet filters properties cannot be configured in the **New IP Packet Filter Wizard**. You will probably want to review each new filter using the **Filter Properties** dialog box to review the properties and see if anything needs to be added to the filter definition.

Co-Hosting IIS on a Computer with ISA Server

In most cases, it is desirable to run ISA Server on a dedicated computer. Small organizations, however, may find it unfeasible to allocate an entire computer's resources to a single service. Apart from performance issues, there is no reason why the ISA Server computer cannot support other services. A common candidate to inhabit a server with ISA Server is IIS, but there are special issues when a Web server is co-hosted with an IIS Server.

The big issue is ports. No two services can use the same port number on the same computer. Outgoing requests do not present a problem because, for example, the Web Proxy service typically listens for outgoing HTTP requests on port 8080 and therefore does not conflict with the use of port 80 for the Web service on IIS.

For incoming Web requests, however, some extra work is required. The chief problem is that ISA Server must offer listeners for HTTP, HTTPS, FTP, and Gopher to outside Web clients. In nearly all cases, the protocols are supported with their default ports. If ISA Server uses port 80 to accept inbound HTTP service requests, any Web server, such as IIS, running in the same server cannot also use port 80.

The procedure used to resolve these port conflicts has two parts:

- Configure IIS to operate its services using non-standard ports. For example, the default Web server could be configured to use port 50080 instead of the standard HTTP port of 80.
- Configure Web publishing and routing rules to accept requests sent to standard ports and to redirect those requests to the IIS server using the non-standard port that has been assigned.

There is always a risk when using non-standard ports. You don't want to step on a port that is already being used, and you don't want to install new software that expects to use the port without reconfiguring either the new software or IIS. At the end of the section "All Ports Scan Probes," a note discusses port scanners. A port scanner such as the AW Security Port Scanner is an effective means of learning the ports that are in use on a computer.

It is preferable not to use a port that IANA has assigned for another protocol. Avoid using well-known ports (0-1043), most of which are assigned to high-profile protocols. Some secondary ports are specified in the registered ports range (1024 through 49151.) For example, 8080 is registered as an alternative HTTP port. There are, however, no registered secondary

ports for HTTPS, FTP, or Gopher, so proceed with caution. Finally, addresses in the dynamic (or private) ports range (49152 through 65535) are not registered and can be used for any purpose. It is up to you to ensure that there are no conflicts when selecting ports.

Managing the Cache

Several aspects of the ISA Server cache can be configured: cache properties, cache allocation to drives, and schedules for automated downloads.

Configuring Cache Properties

To open the **Cache Configuration Properties** dialog box, right-click **Arrays**→*array*→**Cache Configuration** and choose **Properties** from the context menu.

Cache Properties: The General Tab

The **General** tab is informational only, reporting the total cache size of all servers in the array.

Cache Properties: The HTTP Tab

Figure 51 shows the **Cache Configuration Properties** dialog box with the **HTTP** tab selected. The option **Enable HTTP caching** must be selected to activate other options in the dialog box.

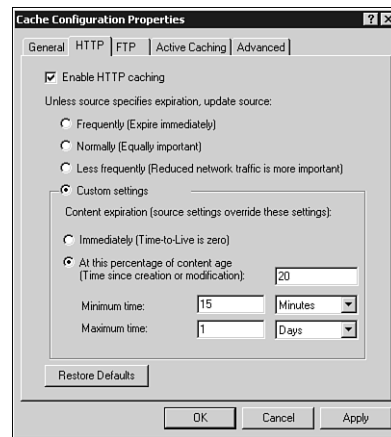


FIGURE 51

Cache Properties: The HTTP tab.

The HTTP cache configuration properties determine how long objects are retained in cache and the frequency with which they are updated. These settings take effect only when the object does not specify a TTL. When an object TTL is reached, the object is expired. Expired objects

may still be provided to clients if the source Web server is unavailable. Otherwise, these settings determine the object currency that ISA Server will attempt to maintain for cached objects.

Four choices are available:

- **Frequently (Expire immediately).** This option forces the cache to expire objects frequently, requiring ISA Server to retrieve the objects more often. Objects provided to clients are as current as possible at the cost of a significant increase in network traffic.
- **Normally (Equally important).** This is a compromise setting that establishes a longer TTL. Objects provided to users may no longer be current, but network traffic is reduced.
- **Less frequently (Reduced network traffic is more important).** This option causes ISA Server to assign a long TTL to objects. Network traffic is significantly reduced, but the likelihood of providing outdated objects in response to client requests increases.
- **Custom settings.** Select this option to define custom HTTP caching behavior.

If **Custom settings** is selected, the following options can be used to specify cache behavior:

- **Immediately (Time-to-Live is zero).** With this setting, ISA Server must obtain a new copy of an object each time a client requests it. This setting nearly eliminates the advantages offered by a cache. Caching is configured for each array, and it is entirely conceivable that you might want to set a TTL of zero for one array and not for another.
- **At this percentage of content age (Time since creation or modification).** The field accepts a value from 0 through 100. A value of 0 results in a TTL of zero and immediate object expiration. Higher values result in progressively longer TTLs and longer intervals before objects are expired. Minimum and maximum TTLs can be established as follows:
 - **Minimum time.** Specify a number of units and a unit duration (seconds, minutes, hours, days, or weeks) to establish a minimum TTL that will be applied to objects. A minimum time that is too short will produce high levels of network traffic.
 - **Maximum time.** Specify a number of units and a unit duration to establish a maximum TTL that will be applied to objects. A maximum time that is unreasonably old may result in failure to expire objects that have changed on the source server and need to be refreshed.

Think of the **percentage of content age** value in this way: A retrieved object that has not changed for a long period of time can be considered to be highly stable and therefore can safely be assigned a long TTL with little risk. An object that was renewed recently may be volatile and should be assigned a short TTL. If an object is modified and then remains stable for a considerable time, its increased age will result in progressively longer TTLs.

Cache Properties: The FTP Tab

The **FTP** tab has two options:

- **Enable FTP caching.** I won't insult you by explaining.
- **Time to live for all objects.** When caching is enabled, FTP applies a "one size fits all" TTL policy. Specify a number of units and a unit duration to establish the TTL that will be applied to objects in the FTP cache.

Cache Properties: The Active Caching Tab

Active caching, when enabled, configures ISA Server to refresh select objects during periods of low demand. Checking **Enable active caching** enables...gee, I'm not sure. Can you figure it out?

When active caching is enabled (that's probably it) three settings determine how objects are retrieved:

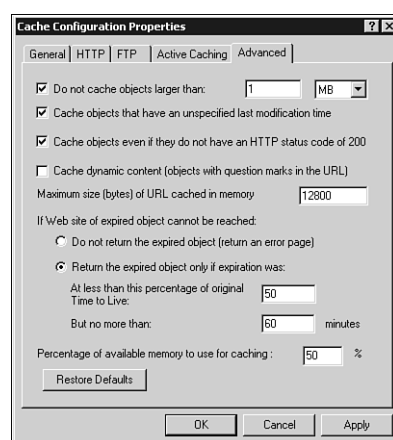
- **Frequently.** Select this option to ensure that a valid copy of the object is in cache, thereby improving client performance at the cost of greater network traffic.
- **Normally.** This setting establishes a compromise between client performance and network traffic.
- **Less frequently.** Network traffic is reduced, although client performance may be reduced as well.

Cache Properties: The Advanced Tab

The **Advanced** tab, shown in Figure 52, includes a potpourri of various cache settings. The first five settings are

- **Do not cache objects larger than.** If this option is enabled, you can define a maximum size limit for cached objects in KB, MB, or GB. To an extent, use of this option is determined by the available cache size. Clearly, you don't want one or a few objects occupying most of the available cache. When storage space is ample, however, it makes sense to cache even large objects since these objects are slow to transfer and burn significant amounts of bandwidth. This option is disabled by default.
- **Cache objects that have an unspecified last modification time.** Objects that do not include a last modification time are cached only if this option is enabled. The header does contain information regarding when the request was received and a time for the request to expire. This option is enabled by default.
- **Cache objects even if they do not have an HTTP status code of 200.** 200 is the HTTP status code that identifies an OK response. It may be useful to cache responses when objects cannot be retrieved if those objects are requested frequently, since caching those responses can still reduce network traffic. The practice of caching negative responses is called *negative caching*. This option is enabled by default.

- **Cache dynamic content (objects with question marks in the URL).** Dynamic content is generated freshly in response to each query and typically each response is unique. In such situations, caching responses is of little or no value. When the same request may be repeated, obtaining the same result, caching may be useful. This option is disabled by default.
- **Maximum size (bytes) of URL cached in memory.** This parameter establishes a maximum size in bytes for URLs that are cached. The default value is 12,800 bytes.

**FIGURE 52**

Cache Properties: The Advanced tab.

If an object in cache is expired and cannot be refreshed because the Web site that returned the object is unavailable, ISA Server can respond to requests for the object in two ways:

- **Do not return the expired object (return an error page).** This option prevents ISA Server from returning a cached object that may be obsolete.
- **Return the expired object only if expiration was.** If this option is selected, expired objects may be returned depending on the settings of the following two parameters:
 - **At least this percentage of original Time to Live.** This parameter specifies the maximum time that an expired object will be returned where the expired time is a percentage of the original TTL. If this parameter is 50, the object will be returned until it has been expired for 50% of the original TTL, that is to say, until the object's age reaches 150% of its original TTL. The default value is 50.
 - **But no more than.** This parameter specifies a maximum age for objects that will be returned in terms of an absolute age limit in minutes. The object will be returned until its age reaches the lesser of the times established by the two formulas.

Finally, the **Percentage of available memory to use for caching** parameter prevents caching from preempting all available RAM. Of course, this percentage can be higher on a dedicated ISA Server with plenty of memory than on a non-dedicated ISA Server that has limited memory.

Configuring Cache Disk Size

When you select **Arrays**→**array**→**Cache Configuration**→**Drives**, the Details pane lists each ISA Server in the array along with the sizes of available disks on the server and the size of the cache that is configured.

To change the cache size on a server:

1. Right-click a server in the Details pane and choose **Properties** in the context menu.
2. Select a drive in the drive list.
3. Specify the cache size for that drive in the **Maximum cache size (MB)** field.

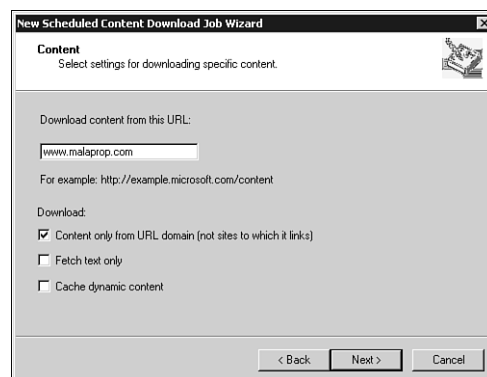
Scheduling Content Downloads

If you want to ensure that specific content is always available in cache, you can schedule a content download job. Configured jobs are listed in the Details pane when you select **Arrays**→**array**→**Cache Configuration**→**Scheduled Content Download Jobs**.

To schedule a new job:

1. Right-click **Scheduled Content Download Jobs** and choose **New**→**Job** in the context menu to start the **New Scheduled Content Download Job Wizard**.
2. In the **Welcome** dialog box, enter a name for the job.
3. In the **Start Time** dialog box, enter a date and time when the job is to be executed, or, for repeating jobs, the first time the job is to be executed.
4. In the **Frequency** dialog box, select one of these choices:
 - **Once**. The job will not recur.
 - **Daily**. The job is executed daily at the time specified in the **Start Time** dialog box.
 - **Weekly on**. When this option is selected, check the days on which the job is to occur. The job will be executed every week on the days specified, at the time specified in the **Start Time** dialog box.
5. Complete the **Content** dialog box (Figure 53) as follows:
 - **Download content from this URL**. Supply the URL to which the request is directed.
 - **Content only from URL domain (not sites to which it links)**. Enable this option unless you know the extent to which including links will extend the size of the response. In some cases, links can pull in an overwhelming amount of content.

- **Fetch text only.** Select this option to omit non-text content such as graphics and sound from the response.
- **Cache dynamic content.** As discussed previously, dynamic content often is a bad candidate for caching. Enable this option or not based on your examination of the content and of the frequency of requests for that content.

**FIGURE 53**

Specifying the content for a scheduled content download job.

6. The **Links and Downloaded Objects** dialog box appears next, as shown in Figure 54. This dialog box includes several types of settings:

TTL settings are

- **Always override object's TTL.** Select this option if you want the TTLs of all retrieved objects set to a value you specify.
- **Override TTL if not defined.** The TTL will be overridden only if a TTL is not specified in the object's header.
- **Mark downloaded objects with a new TTL of.** Specify the TTL in minutes that is to be applied to retrieved objects under the conditions specified by the previous two check boxes.

Links depth settings determine the maximum number of links that can be applied to a cached object:

- **Traverse maximum links depth of.** Specify a link depth that limits the depth of the object search.
- **No limit on maximum depth.** If this option is chosen, objects will be retrieved regardless of the number of links that are required to retrieve them. Use this option with caution. Be sure you know where the links are directed and the amount of data that will be retrieved. This option is a terrific method of filling the cache with unwanted objects.

Number of cached objects determines whether the query response is limited in terms of the number of objects it can contain. Two settings are available:

- **Maximum.** Select this option to specify a maximum number of objects.
- **No limit.** Select this option to retrieve all objects that are returned. Like any unlimited parameter, this one is a good way for things to go haywire. Make some tests before you choose this option.

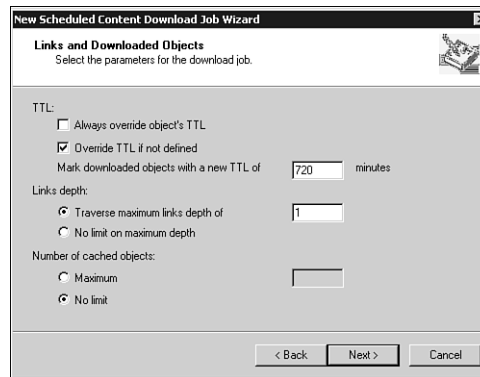


FIGURE 54

Links and downloads properties define the scope of a scheduled download and the manner in which TTLs are assigned to downloaded objects.

Monitoring ISA Server Operation

ISA Server provides a variety of tools for monitoring its operation and alerting administrators of server events. We'll look briefly at them in this section.

Performance Monitor

In the Start Menu container **Start**→**Programs**→**Microsoft ISA Server**, the ISA Server setup program creates an icon named **Monitor Microsoft ISA Server Performance**. Double-click this icon to start Performance Monitor with a wide variety of ISA Server counters. Examples of these counters are

- Disk Cache Allocated Space (KB)
- Active UDP connections
- Active TCP connections
- Active Sessions

- Current Users
- Cache Hit Ratio

These counters can be of tremendous value when performance tuning ISA Server.

Alerts

A large variety of alerts are defined for ISA Server. These alerts are located in the ISA Server console in the container **Arrays**→*array*→**Monitoring Configuration**→**Alerts**, shown in Figure 55. The alerts can be used to notify administrators of ISA Server operations in a variety of ways.

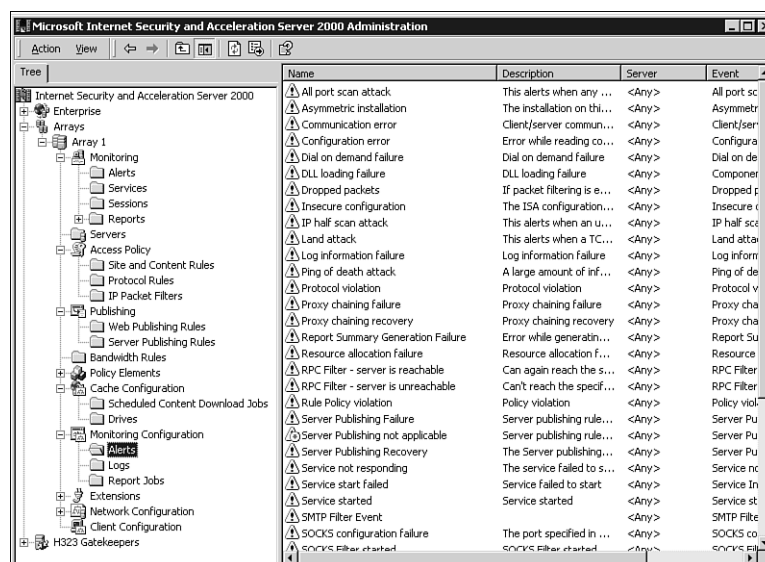


FIGURE 55

Alerts that can be used to monitor ISA Server.

Modify the properties of an alert as follows:

1. Open the alert properties dialog box by right-clicking the alert and choosing **Properties** from the context menu.
2. The **General** tab has three properties:
 - **Name.**
 - **Description.**
 - **Enable.** Check to enable the alert, remove the check to disable the alert.

3. The **Events** tab is shown in Figure 56. Not all fields are active for all alerts. Most alerts are pre-configured with **Events** properties that are appropriate in type and value for the type of alert. The **Events** tab has the following options:

- **Additional condition.** For some alerts additional conditions can be imposed. For example, the alert might be tied specifically to the Web Proxy service. If desired, select conditions from the drop-down list.
- **By server.** Some alerts can be applied to all servers in the array or to specific servers.
- **Number of occurrences before the alert is issued.** For some alerts, it often makes sense to issue an alert only if the condition has recurred a minimum number of times. For example, an **All ports scan attack** alert might be required to repeat 10 times before notifying an administrator.
- **Number of events per second before the alert is issued.** Many conditions are of concern only when they occur with too great a frequency. This option can be used to specify a frequency in events-per-second that will trigger the alert.
- Three options determine when an alert action is generated with regard to an alert exception:
 - **Immediately.** The action is triggered immediately after the exception occurs.
 - **After manual reset of alert.** The action is triggered after the alert is reset following an exception.
 - **If the time since last execution is more than n minutes.** With this option, the action takes place at intervals of n minutes.

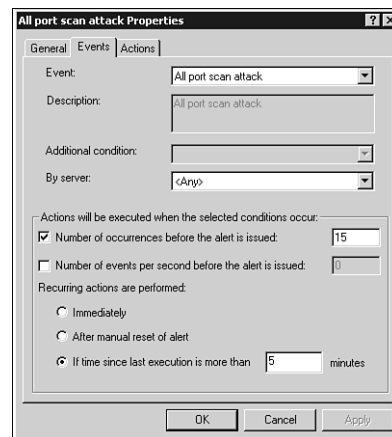


FIGURE 56

Configuring alert events.

4. The **Actions** tab is shown in Figure 57. Actions are the ways ISA Server responds to alert exceptions. The **Actions** tab has the following fields:
- **Send e-mail.** When this box is checked, ISA Server will send an email via the SMTP mail server specified to the mail accounts that are noted in the **To** and **CC** fields.
 - **Program.** When this box is checked, ISA Server will execute the program you specify.
 - **Report to Windows 2000 event log.** This is the only action that is enabled by default.
 - **Stop selected services** and **Start selected services.** The following services can be stopped and started:
 - Firewall service
 - Web Proxy service
 - Scheduled cache content download service

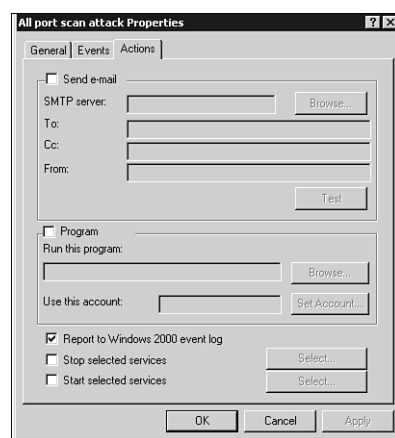


FIGURE 57
Configuring alert actions.

Monitoring Alerts, Services, and Sessions

Select **Arrays**→**array**→**Monitoring**→**Alerts** to access three useful monitoring tools. If you are in Taskpad view, tabs at the bottom of the Taskpad can be used to select tools to monitor alerts services and sessions.

Figure 58 shows the **Monitor Servers and Services** Taskpad. Listed in the Taskpad are all servers in the array with all services running on each server. You can easily stop, start, and determine the status of each service.

The **Monitor Alerts** Taskpad lists all alerts that have been triggered. Alerts can be reset from the Taskpad. This is one of the most useful screens to leave up on the ISA Server monitor.

The **Monitor Sessions** Taskpad lists all active sessions. Sessions can be disconnected from this Taskpad.

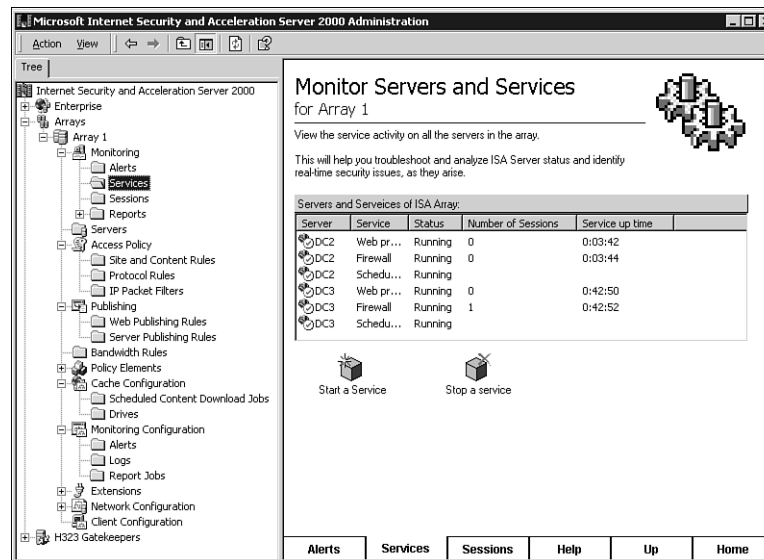


FIGURE 58

Monitoring ISA Servers and services.

Configuring ISA Server Clients

Some configuration is required to enable clients to take advantage of ISA Server features. Firewall and Web Proxy clients in particular require proper setup to access their respective ISA Server services.

Configuring SecureNAT Clients

SecureNAT clients require no special configuration. Enabling these clients to use ISA Server is a simple matter of configuring the ISA Server as the clients' default gateway, or ensuring that routing is in place that will direct client queries for outside services through the ISA Server.

SecureNAT clients always route requests to an IP address. They cannot address packets to the array to take advantage of load balancing.

Configuring Firewall Clients

Firewall clients run the Firewall client software, which is installed on the client from a share on the ISA Server. After the Firewall client is installed, some configuration is required at the server and client end. Two steps are required when configuring Firewall clients:

- Configuring Firewall client properties on the ISA Server
- Installing and configuring the Firewall client software on the client computer

Configuring ISA Server Support for Firewall Clients

The first step in configuring ISA Server for Firewall clients is to configure the Firewall client properties as follows:

1. Click **Arrays**→*array*→**Client Configuration**.
2. In the Details pane, right-click **Firewall Client** and choose **Properties** in the context menu.
3. Select one of the following options to be used to configure the Firewall client when it is being installed:
 - If the ISA Server or array is identified by a Host Address RR in DNS, select **Array DNS name** and enter the fully qualified domain name of the server or array in the field that is provided.
 - If the ISA Server or array is identified by an IP address, select **IP address** and enter the IP address in the field that is provided.

The settings entered in this dialog box will be used to configure the Firewall client software when it is installed on the client computer.

NOTE

It is often useful to assign DNS domain names to ISA Server arrays. These are simply Host Address RRs that map a hostname to the internal IP address of a standalone ISA Server or to the internal IP addresses of all ISA Servers in an array.

Installing and Configuring the Firewall Client Software

The Firewall client software is installed on the ISA Server computer with the sharename `mspc1nt`. To install the Firewall client, on the client computer, connect to the `mspc1nt` share on an ISA Server in the array the client will use. Then execute `Setup.exe` in the shared folder. The only decision to be made during installation is whether or not to use the default installation folder.

The Firewall client is configured through a **Firewall Client** applet that is added to the Control Panel. The **Firewall Client Options** dialog box is extremely simple, as Figure 59 illustrates, with the following options:

- **Enable Firewall Client.** Check to enable the firewall client.
- **Automatically detect ISA Server.** It is disabled in the figure. We will look at the procedures for configuring automatic ISA Server detection later in the section “Configuring Automatic Server Discovery.”
- **Use this ISA Server.** Specify the name of the ISA Server the client is to use. If DNS maps this name to the internal IP addresses of the ISA Servers in an array, round-robin addressing distributes usage across all array servers.
- **Update Now.** Click this button to connect to the Firewall service. If the Firewall Client properties on the ISA Server are properly configured, the client will refresh its configuration from the Firewall service and a box will present the message **Refresh Operation Completed Successfully**.

The Firewall client can now communicate through the Firewall service.



FIGURE 59

Options for configuring the Firewall client.

Firewall clients receive configuration information in the form of two files named `mssc1nt.ini` and `wspcfg.ini`, which are copied to the client when it connects to the server and are downloaded periodically thereafter. In some cases, it may be necessary to manually configure these files. See the topic “Advanced firewall client configuration” in the ISA Server Help for information.

CAUTION

Do not install the Firewall client software on an ISA Server computer.

Configuring Web Proxy Clients

Web Proxy clients are configured in two stages:

- Web Proxy browser configuration support must be configured on the ISA Server.
- Web Proxy clients must be configured.

Web Browser Configuration Support

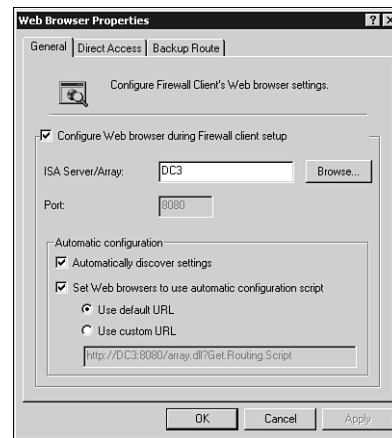
Before enabling a lot of Web Proxy clients, take the time to configure Web Browser properties on the ISA Server. These properties can greatly simplify browser configuration.

To configure Web Browser properties, do the following:

1. Select **Arrays**→*array*→**Client Configuration**.
2. In the Details pane, double-click **Web Browser** to open the **Web Browser Properties** dialog box.
3. The **General** tab is shown in Figure 60. This tab has the following properties:
 - **Configure Web browser during Firewall client setup.** If this option is enabled, Windows Explorer version 5 will be configured automatically when the Firewall client software is installed. Other Web browsers cannot be configured in this manner.
 - **ISA Server/Array.** Enter the Active Directory or DNS FQDN for the ISA server, or the FQDN for the ISA Server array.
 - **Port.** Enter the port that internal Web Proxy clients use to send outgoing Web requests to the Web Proxy service. In many cases, this setting is already established for the array on the **Outgoing Web Requests** dialog box of the **Array Properties** dialog box. The standard port is 8080.
 - **Set Web browsers to use automatic configuration script.** Select this option if Web browsers are to be automatically configured from a script. Then select one of the following options to specify the location of the script:
 - **Use default URL.** The Web Proxy service maintains a default configuration script that is suitable for use in most cases. To use the default script, select this option.
 - **Use custom URL.** Custom Web browser configuration scripts can be defined, in which case the URL of the custom script must be entered. The URL that appears in the text field is actually the URL for the default configuration script.

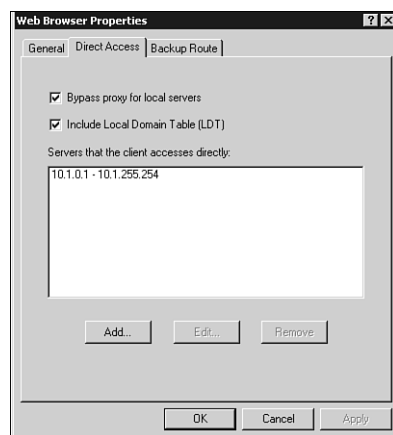
NOTE

Windows Explorer versions 3.02 and later and Netscape versions 2.0 and later can be configured by these scripts.

**FIGURE 60**

Web Browser Properties: The General tab.

- The **Direct Access** tab is shown in Figure 61. These properties describe the servers that clients will communicate with directly, rather than through the Web Proxy. Complete the properties on this tab as follows:
 - **Bypass proxy for local servers.** Check this option if Web Proxy clients will access local Web servers directly, without routing requests through ISA Server.
 - **Include Local Domain Table (LDT).** Check this option if the client is to receive a copy of the LDT.
 - **Servers that the client accesses directly.** Entries of one or more servers can be added to this list. Entries can consist of a single IP address, a range of IP addresses, or the domain name suffix of a domain that is considered to be local. (Web Proxy clients do not use the LAT. Local addresses must be separately specified.)

**FIGURE 61**

Web Browser Properties: The Direct Access tab.

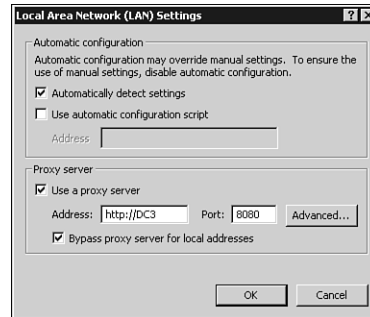
5. The **Backup Route** tab can be used to define a backup route that will be used if the Web Proxy client's primary Web Proxy server is unavailable. To enable support for a backup route, check **If ISA Server is unavailable, use this backup route to connect to the Internet**. Then choose one of the following options:
 - **Direct access.** Select this option if the client can access the Internet through a conventional router or a network that connects directly to the Internet.
 - **Alternative ISA Server.** Select this option if the client should redirect requests to another ISA Server. Specify the IP address, AD name, or domain name of the other ISA Server.

Configuring the Web Proxy Client

In most cases, "Web Proxy Client" means a Web Proxy-enabled browser such as Windows Explorer. Here is the procedure to configure the Web Proxy client in Internet Explorer 5:

1. Start Internet Explorer.
2. Select **Tools**→**Internet Options** in the menu bar to open the **Internet Options** dialog box.
3. Select the **Connections** tab and click **LAN Settings** to open the **Local Area Network (LAN) Settings** dialog box, shown in Figure 62. Configure the proxy settings as follows:
 - **Automatically detect settings.** Enable this option if the client should attempt to discover its Web Proxy server settings.
 - **Use Automatic configuration script.** If the Web Proxy server offers an automatic configuration script, as described in the previous section, enter the URL for the script in the space provided.

- **Use a proxy server.** Select this option if you want to manually specify the client's Web Proxy server. Other fields in the **Proxy server** box are enabled only when this option is enabled. Describe the proxy server with the following properties:
 - **Address.** Identify the Web Proxy server by entering its URL, name, or IP address.
 - **Port.** The default port for sending requests to the proxy server is 8080. By default, this port is used to receive all Web-based service requests.
 - **Bypass proxy server for local addresses.** If this option is selected, the browser will communicate directly with Web servers that are identified in the LDT or in the table of local IP addresses.

**FIGURE 62**

Configuring Internet Explorer 5 as a Web Proxy client.

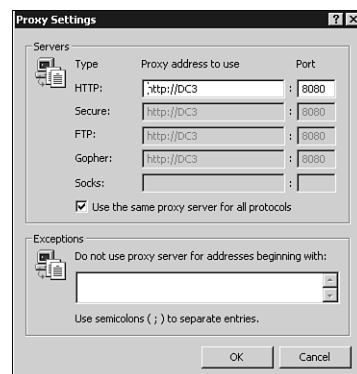
4. To configure additional properties, click the **Advanced** button to open the **Proxy Settings** dialog box shown in Figure 63.

Properties in the **Servers** box enable you to configure proxy settings individually for each Web protocol. Initially, **Use the same proxy server for all protocols** is selected, and all services are configured to use the same proxy server and port.

To configure servers separately, disable **Use the same proxy server for all protocols**. You will then be permitted to edit the server addresses and port numbers individually.

An individual list of local IP addresses can be created in the **Exceptions** list box.

The client is now configured to direct Web service requests destined for external Web servers to the proxy server for forwarding.

**FIGURE 63**

Configuring Internet Explorer 5 advanced proxy settings.

Configuring Automatic Server Discovery

Web Proxy and Firewall clients can be given the abilities to automatically locate their preferred Web Proxy and Firewall servers. Discovery is performed by way of special records in DNS or DHCP.

Enabling Automatic Server Discovery for Web Proxy Clients

Automatic discovery is supported for Internet Explorer 5.0 and later versions. For clients that are configured by a script obtained from ISA Server, do the following in the ISA Server console:

1. Select **Arrays**→**array**→**Client Configuration** and double-click **Web Browser** in the Details pane to open the **Web Browser Properties** dialog box.
2. On the **General** tab, enable **Automatically discover settings**.

To configure automatic Web Proxy server discovery on clients that are not configured by the Web Proxy script, start Internet Explorer and do the following:

1. Select **Tools**→**Internet Options**→**Connections**→**LAN Settings** to open the **Local Area Network (LAN) Settings** dialog box.
2. Enable **Automatically detect settings**.

Enabling Automatic Server Discovery for Firewall Clients

For clients that are configured by a script obtained from ISA Server, do the following in the ISA Server console:

1. Select **Arrays**→*array*→**Client Configuration** and double-click **Firewall Client** in the Details pane to open the **Firewall Client Properties** dialog box.
2. On the **General** tab, check **Enable ISA Firewall automatic discovery in Firewall Client**.

To enable automatic ISA Server discovery on firewall clients:

1. Open the **Firewall Client** applet in the Control Panel.
2. Enable **Automatically detect ISA server**.

Configuring DNS for Automatic Server Discovery

Clients that are set up for automatic ISA Server discovery first request an object from an ISA Server that is configured to service requests. If the ISA Server does not respond, the client attempts to send WPAD or WSPAD requests to DNS and/or DHCP. Both DNS and DHCP must be configured to respond to the requests.

The procedure for configuring DNS to support automatic discovery is as follows:

1. Start the **DNS** console and open the domain that contains the ISA Server array members.
2. Create a Host Address RR for a standalone ISA Server. For an array, create a Host Address RR that matches the array name to each member of the array so that round-robin addressing is supported. The hostname need not be identical to the name of the array in the ISA Server console, and cannot be if the array name includes spaces. A useful approach is to substitute underscores for spaces in the hostname.
3. To locate the Web Proxy service, create an Alias (CNAME) RR. In **Alias Name** enter WPAD. In **Fully qualified name for target host** type the FQDN of the ISA Server or server array.

It is not necessary to define a WSPAD entry. ISA Server consults the WPAD entry for the information required to define a WSPAD entry.

When a client that has automatic detection needs to send a service request, it looks in DNS or DHCP for a WPAD or WSPAD entry. This entry enables the client to communicate with the ISA Server that will process the request.

Configuring DHCP for Automatic Server Discovery

The procedure for configuring DNS to support automatic discovery by Web Proxy clients is as follows:

1. Start the **DHCP** console, right-click the desired DHCP server in the object tree, and choose **Set Predefined Options** to open the **Predefined Options and Values** dialog box.
2. Click **Add** to open the **Option Type** dialog box.
3. Define properties for the new option type as follows:
 - **Name.** WPAD
 - **Data Type.** String
 - **Code.** 252
4. Click **OK** to return to the **Predefined Options and Values** dialog box.
5. For **Value** enter `http://string/wpad.dat` where *string* has one of the following values:
 - WPAD if a DNS server is configured to resolve WPAD queries.
 - The FQDN of an ISA Server or array if WPAD queries are not supported by a DNS server.
6. Right-click **Server Options** or **Scope Options**.
7. Check option 252 and complete the **Value** field as described in Step 5.

When a client that has automatic detection needs to send a service request, it looks in DNS or DHCP for a WPAD or WSPAD entry. This entry enables the client to communicate with the ISA Server that will process the request.

Safe and (More) Secure

No security solution can be trusted to be bulletproof, but ISA Server can significantly reduce the vulnerability of your network. It is possible to purchase more secure, more flexible, and better-performing firewalls than ISA Server, and if your organization can budget for a third-party firewall, by all means go for it. But the Microsoft option offers a lot of bang for the buck and lets you administer a firewall using familiar Windows 2000 Server techniques.

Just don't "set it and forget it." Try to penetrate your network. Better yet, hire an expert to try to penetrate your network and possibly to help you screw things down a bit tighter. Monitor the logs, use alerts to notify you when something potentially nasty is happening, and keep current on security announcements posted on Microsoft's Web site. Security isn't an accomplishment, it's a state of mind.